

**A FRAMEWORK FOR THE DETERMINATION OF
WEAK PARETO FRONTIER SOLUTIONS UNDER
PROBABILISTIC CONSTRAINTS**

A Thesis
Presented to
The Academic Faculty

by

Hongjun Ran

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in Aerospace Engineering

Georgia Institute of Technology

May 2007

Copyright © 2007 by Hongjun Ran

**A FRAMEWORK FOR THE DETERMINATION OF
WEAK PARETO FRONTIER SOLUTIONS UNDER
PROBABILISTIC CONSTRAINTS**

Approved by:

Professor Dimitri Mavris, Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Professor Daniel Schrage
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Michelle Kirby
School of Aerospace Engineering
Georgia Institute of Technology

Professor Kwok-Leung Tsui
School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor Vitali Volovoi
School of Aerospace Engineering
Georgia Institute of Technology

Date Approved: April 5, 2007

ACKNOWLEDGEMENTS

No words can express my gratitude enough to my advisor, Dr. Mavris. I could not have attained the accomplishment of this research if he would not have given his wise and priceless guidance and comments for me to go along the right direction, form the right methods, and do the right experiments. I have learnt so much from him, and still have so much more to learn from him. It is a tremendous endeavor to support a student towards a Ph.D. degree. Without his generosity for such a long course of study, I would not have this opportunity to reach this new stage of my life. He will always have my respect and gratitude.

I would like to thank Drs. Schrage, Kirby, Tsui, and Volovoi for their support as committee members, reviewing my thesis, and for their feedback. Special thanks go to Dr. Kirby for many enlightening discussions and comments that shaped the outline of my thesis. Reid Thomas and Dr. Janel Nixon are to be thanked for the fruitful comments and editorial review of my thesis with patience. I also wish to thank my fellow peers and friends who have helped me during the course of my Ph.D. study, either with the research projects or my thesis.

Thank you all.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	iii
LIST OF TABLES	viii
LIST OF FIGURES	ix
LIST OF SYMBOLS AND ABBREVIATIONS	xiii
SUMMARY	xx
1 INTRODUCTION AND MOTIVATION.....	1
1.1 The Conceptual Design Process.....	1
1.2 The Design Process Paradigm Shift and Required Design Methods	4
<i>1.2.1 The Design Process Paradigm Shift and Its Implications</i>	<i>4</i>
<i>1.2.2 Concepts of Multi-Objective Optimization Methods</i>	<i>9</i>
<i>1.2.3 Concepts of Multidisciplinary Optimization Methods</i>	<i>11</i>
<i>1.2.4 Concepts of Probabilistic Design Methods.....</i>	<i>12</i>
1.3 The Need for a New Framework for Design Alternative Generation and Selection ...	16
1.4 Research Objective.....	18
2 BACKGROUND	20
2.1 Mathematical Statement of Realistic Conceptual Design Problems and Solving Considerations	20
2.2 Sampling Methods / Design of Experiments	24
<i>2.2.1 Three Modern DoE's</i>	<i>24</i>
2.3 Surrogate-Modeling Methods.....	31
<i>2.3.1 Surrogate-Modeling Preliminaries.....</i>	<i>31</i>
<i>2.3.2 Two State-of-the-Art Surrogate-Modeling Methods</i>	<i>38</i>
<i>2.3.3 Comparisons of Surrogate-Modeling Methods</i>	<i>57</i>
2.4 Model Assessment and Model Selection Methods	61
<i>2.4.1 Concepts of Model Assessment and Model Selection</i>	<i>61</i>
<i>2.4.2 Information Criteria</i>	<i>63</i>

2.4.3 <i>Practical Selection of Three SVR Parameters</i>	66
2.5 Preliminaries of Probabilistic Design Methods	69
2.5.1 <i>The Basics of Probabilistic Design</i>	69
2.5.2 <i>Mathematical Foundation of Joint Probability Assessment</i>	75
2.5.3 <i>Joint Probabilistic Assessment Methods</i>	77
2.5.4 <i>Summary of Robust Design and Reliability Design Methods</i>	81
2.6 Multi-Objective and Multidisciplinary Optimization Methods	86
2.6.1 <i>Multi-Objective Optimization Methods</i>	86
2.6.2 <i>Multidisciplinary Optimization Methods</i>	92
3 RESEARCH QUESTIONS AND HYPOTHESES	95
3.1 <i>About Formulation of the New Framework</i>	95
3.2 <i>Research Questions</i>	102
3.2 <i>Hypotheses</i>	105
4 FRAMEWORK FORMULATION	108
4.1 <i>Surrogate Modeling and Model Selection</i>	109
4.1.1 <i>The Space Filling Sampling Methods</i>	109
4.1.2 <i>The Ranges of the Design Variables</i>	110
4.1.3 <i>Normalization of Values of Both Design Variables and Responses</i>	111
4.1.4 <i>The Random Cross validation Method</i>	113
4.1.5 <i>The Modified AIC and BIC Information Criteria</i>	115
4.1.6 <i>Model Selection and the Model Selection Advisor</i>	116
4.1.7 <i>The Scheme for Hybrid Surrogate-Modeling with RSM and SVR and the Model Selection Advisor</i>	118
4.1.8 <i>The Levels of Surrogate Models</i>	120
4.2 <i>Determination of the WPF Solutions under PC's</i>	121
4.2.1 <i>Defining the Neighborhoods for Searching Consistent Designs</i>	122
4.2.2 <i>Two Schemes for the Neighborhood Search Method</i>	128
4.2.3 <i>Relaxation of Converging Conditions for Coupling Variables and Thresholds in PC's</i>	133
4.2.4 <i>The Flowchart for Determination of the WPF Solutions of a JPMOMDO Problem</i>	134
5 IMPLEMENTATION AND RESULTS	137

5.1 Pure Mathematical Examples of Surrogate Modeling	137
5.1.1 <i>The Upper Hemisphere Example</i>	138
5.1.2 <i>The Wave Function Example</i>	143
5.1.3 <i>The Rastrigin Function with 36 Peaks Example</i>	149
5.1.4 <i>Comparison with Neural Network</i>	154
5.1.5 <i>Discussion</i>	159
5.2 Pure Mathematical Examples of Finding the Weak Pareto Frontier	161
5.2.1 <i>First Mathematical Example of Finding WPF</i>	162
5.2.2 <i>Second Mathematical Example of Finding WPF</i>	164
5.2.3 <i>Third Mathematical Example of Finding WPF</i>	166
5.2.4 <i>Fourth Mathematical Example of Finding WPF</i>	168
5.2.5 <i>Discussion</i>	171
5.3 A Transport Aircraft Design Example	172
5.3.1 <i>Surrogate Models</i>	174
5.3.2 <i>Design Results</i>	176
5.3.3 <i>Discussion</i>	184
5.4 A Reusable Launch Vehicle Design Example	187
5.4.1 <i>Surrogate Models</i>	189
5.4.2 <i>Design Results</i>	191
5.4.3 <i>Discussion</i>	196
6 CONCLUSIONS AND RECOMMENDATIONS.....	199
6.1 <i>Research Questions</i>	199
6.2 <i>Summary of Contributions</i>	205
6.3 <i>Recommendations</i>	208
APPENDIX A: SAMPLING METHODS OVERVIEW AND SOME MODERN METHODS	211
A.1 <i>Overview of Sampling Methods</i>	211
A.2 <i>Overview of Classical DoE's</i>	215
A.3 <i>Orthogonal Array Sampling</i>	216
A.4 <i>Uniform Designs</i>	218

APPENDIX B: SURROGATE-MODELING PRELIMINARIES AND THREE MODERN METHODS.....	220
<i>B.1 Statistical Inferences</i>	<i>220</i>
<i>B.2 The Problem of “Curse of Dimensionality” of the Parametric Inference.....</i>	<i>222</i>
<i>B.3 Problem of Regression Estimation and Related Decision Principles</i>	<i>223</i>
<i>B.4 Neural Network</i>	<i>231</i>
<i>B.5 Gaussian Process.....</i>	<i>236</i>
<i>B.6 Kriging</i>	<i>238</i>
APPENDIX C: USING KARUSH-KUHN-TUCKER CONDITIONS TO CALCULATE b IN SVR.....	242
APPENDIX D: ERROR MEASURES AND TWO MODEL ASSESSMENT AND SELECTION METHODS.....	246
<i>D.1 Model fitting Error and Predicting Error</i>	<i>246</i>
<i>D.2 Fundamental of the Re-Sampling Methods: the Jackknife Method</i>	<i>247</i>
<i>D.3 The Cross Validation Method</i>	<i>249</i>
<i>D.4 The Bootstrap Method</i>	<i>251</i>
APPENDIX E: THE CONCEPTS OF MPP AND LSF OF FPI	253
APPENDIX F: THE MULTI-VARIATE MONTE CARLO SAMPLING	255
APPENDIX G: TRUNCATED NORMAL DISTRIBUTION	257
APPENDIX H: A TRANSPORT AIRCRAFT DESIGN OPTIMIZATION PROBLEM	259
APPENDIX I: A REUSABLE LAUNCH VEHICLE DESIGN OPTIMIZATION PROBLEM	267
REFERENCES.....	275
VITA.....	285

LIST OF TABLES

Table 1: Some Terms of the Dual Optimization Problem for Different Loss Functions.....	54
Table 2: Common Kernel Functions of SVR	56
Table 3: Qualitative Comparison of Surrogate-Modeling Methods	60
Table 4: Values of General Parameters and Goodness of Fit for the Upper Hemisphere	139
Table 5: Values of General Parameters and Goodness of Fit for the Wave Function.....	145
Table 6: Values of General Parameters and Goodness of Fit for the Rastrigin Function with 36 Peaks	150
Table 7: Goodness of Fit of RSSVR and NN for Three Pure Mathematical Examples	155
Table 8: Objective Functions and Features of the Mathematical Examples of Finding WPF	161
Table 9: Exact Single-Objective Deterministic Optimal Results of the Aircraft Design Example.....	173
Table 10: Values of General Parameters and Goodness of Fit for the D CA	175
Table 11: Values of General Parameters and Goodness of Fit for the A CA	175
Table 12: Values of General Parameters and Goodness of Fit for the W CA	175
Table 13: Values of General Parameters and Goodness of Fit for the P CA	176
Table 14: Valid Solutions Closest to Single-Objective Deterministic Optimal Solutions with the First Search Scheme and Surrogate Models for the Aircraft Design Example	178
Table 15: Valid Solutions Closest to Single-Objective Deterministic Optimal Solutions with the Second Search Scheme and Surrogate models for the Aircraft Design Example	180
Table 16: Exact Single-Objective Deterministic Optimal Results of the RLV Design Example	189
Table 17: Values of General Parameters and Goodness of Fit for the P CA	190
Table 18: Values of General Parameters and Goodness of Fit for the T CA	190
Table 19: Values of General Parameters and Goodness of Fit for the Wb CA.....	191
Table 20: Values of General Parameters and Goodness of Fit for the S CA.....	191
Table 21: Valid Solutions Closest to Single-Objective Deterministic Optimal Solutions with the First Search Scheme and Surrogate Models for the RLV Design Example	192
Table 22: Valid Solutions Closest to Single-Objective Deterministic Optimal Solutions with the Second Search Scheme and Surrogate Models for the RLV Design Example	193

LIST OF FIGURES

Figure 1-1: The Design Space and a Design Alternative of a Design Concept	3
Figure 1-2: Design Process of Korean Trainer T-50	4
Figure 1-3: Design Process of a Wing Spar	4
Figure 1-4: "Cost-Knowledge-Freedom" Interaction and Shift for Future Design Paradigm.....	6
Figure 1-5: Pareto Frontier Points and Inefficient Points in the 2D Objective Space.....	7
Figure 1-6: Variable Fidelity of Aircraft Synthesis and Sizing	8
Figure 1-7: Constraints of Deterministic and Probabilistic Design Approaches	14
Figure 1-8: Distributions of a Design Concept and a Design Alternative in Probabilistic Design	15
Figure 2-1: Potential Process of a Realistic Conceptual Design in a Nesting Loop Approach	23
Figure 2-2: Example of Two Dimensional Latin Hypercube Sampling.....	26
Figure 2-3: Example of Two Dimensional Hammersley Sequence Sampling	29
Figure 2-4: Univariate Monte Carlo Sampling Process	30
Figure 2-5: Example to Show the VC Dimension of a Linear Function Family in the 2 dimensional Input Space.....	44
Figure 2-6: Example to Show the One Commonality between Classification and Regression	46
Figure 2-7: Illustrations of Four Loss Functions for SVR	47
Figure 2-8: Example to Account for Slack Variables for Linear Regression with the ϵ -Insensitive Loss Function	50
Figure 2-9: Framework for Sources of Uncertainty and Error.....	70
Figure 2-10: Uncertainty Classification and Design Domains.....	72
Figure 2-11: Reliability versus Robustness on Probability Density Functions	73
Figure 2-12: Illustrations of Joint, Marginal and Conditional Probability Density Functions	76
Figure 2-13: Example Using Empirical CDF to Estimate P_f	79
Figure 2-14: Difference between a Robust Design Solution and a Deterministic Design Solution.....	83
Figure 2-15: Example Implementing Robust Design	83
Figure 2-16: Difference between a Reliability Design Solution and a Deterministic Design Solution	85
Figure 2-17: Example Implementing the Simplified Reliability Design	86
Figure 2-18: Example to Show the Difference between PF and WPF in the 2D Objective Space.....	88
Figure 2-19: Example to Show the difference between PF and WPF in the 3D Objective Space	88
Figure 2-20: Illustration of the Basic Idea of the Goal Attainment Method	90
Figure 2-21: Example of the DSM of a Multidisciplinary Aircraft Design Problem.....	93
Figure 4-1: Illustration of the Difference between the Design Space and Extended Design Space	111
Figure 4-2: Scheme of Random Cross Validation for Estimation of Model Predicting Error	114

Figure 4-3: Scheme for Hybrid Surrogate-Modeling with RSM and SVR and Model Selection	
Advisor - I	119
Figure 4-4: Scheme for Hybrid Surrogate-Modeling with RSM and SVR and Model Selection	
Advisor - II	120
Figure 4-5: Example to Show the Problems with Grid Search	124
Figure 4-6: Illustration of the Neighborhood Search Method	125
Figure 4-7: First Scheme for the Neighborhood Search Method	130
Figure 4-8: Idea of the Second Search Scheme	131
Figure 4-9: Second Scheme for the Neighborhood Search Method	132
Figure 4-10: Pseudo Program for Relaxation of Convergence Conditions of Coupling Variables	133
Figure 4-11: Pseudo Programs for Relaxation of Constraint Conditions in PC's	134
Figure 4-12: Flowchart for Determination of the WPF Solutions under PC's	136
Figure 5-1: Illustration of the Upper Hemisphere	139
Figure 5-2: Surrogate Model for the Upper Hemisphere by RSM – Case 1, 2, 3	140
Figure 5-3: Surrogate Model for the Upper Hemisphere by SVR – Case 1	140
Figure 5-4: Surrogate Model for the Upper Hemisphere by Hybrid – Case 1	141
Figure 5-5: Surrogate Model for the Upper Hemisphere by SVR – Case 2	141
Figure 5-6: Surrogate Model for the Upper Hemisphere by Hybrid – Case 2	142
Figure 5-7: Surrogate Model for the Upper Hemisphere by SVR – Case 3	142
Figure 5-8: Surrogate Model for the Upper Hemisphere by Hybrid – Case 3	143
Figure 5-9: Illustration of the Wave Function	144
Figure 5-10: Surrogate Model for the Wave Function by RSM – Case 1, 2, 3	145
Figure 5-11: Surrogate Model for the Wave Function by SVR – Case 1	146
Figure 5-12: Surrogate Model for the Wave Function by Hybrid – Case 1	146
Figure 5-13: Surrogate Model for the Wave Function by SVR – Case 2	147
Figure 5-14: Surrogate Model for the Wave Function by Hybrid – Case 2	147
Figure 5-15: Surrogate Model for the Wave Function by SVR – Case 3	148
Figure 5-16: Surrogate Model for the Wave Function by Hybrid – Case 3	148
Figure 5-17: Illustration of the Rastrigin Function with 36 Peaks	150
Figure 5-18: Surrogate Model for the Rastrigin Function by RSM – Case 1, 2, 3	151
Figure 5-19: Surrogate Model for the Rastrigin Function by SVR – Case 1	151
Figure 5-20: Surrogate Model for the Rastrigin Function by Hybrid – Case 1	152
Figure 5-21: Surrogate Model for the Rastrigin Function by SVR – Case 2	152
Figure 5-22: Surrogate Model for the Rastrigin Function by Hybrid – Case 2	153
Figure 5-23: Surrogate Model for the Rastrigin Function by SVR – Case 3	153
Figure 5-24: Surrogate Model for the Rastrigin Function by Hybrid – Case 3	154
Figure 5-25: Surrogate Model for the Upper Hemisphere by ANN – Comparison	156

Figure 5-26: Surrogate Model for the Upper Hemisphere by Hybrid – Comparison	156
Figure 5-27: Surrogate Model for the Wave Function by ANN – Comparison	157
Figure 5-28: Surrogate Model for the Wave Function by Hybrid – Comparison	157
Figure 5-29: Surrogate Model for the Rastrigin Function by ANN – Comparison	158
Figure 5-30: Surrogate Model for the Rastrigin Function by Hybrid – Comparison	158
Figure 5-31: Objective Space of the First Mathematical Example of Find WPF	163
Figure 5-32: Results of the First Mathematical Example of Finding WPF	164
Figure 5-33: Objective Space of the Second Mathematical Example of Find WPF	165
Figure 5-34: Results of the Second Mathematical Example of Finding WPF	166
Figure 5-35: Objective Space of the Third Mathematical Example of Find WPF	167
Figure 5-36: Results of the Third Mathematical Example of Finding WPF	168
Figure 5-37: Objective Space of the Fourth Mathematical Example of Find WPF.....	170
Figure 5-38: Results of the Fourth Mathematical Example of Finding WPF	171
Figure 5-39: WPF Found by the First Search Scheme with 9900 points of S_2 and Surrogate Models for the Aircraft Design Example	177
Figure 5-40: WPF Found by the First Search Scheme with 19800 points of S_2 and Surrogate Models for the Aircraft Design Example	177
Figure 5-41: WPF Found by the Second Search Scheme with 9900 points of S_2 and Surrogate Models for the Aircraft Design Example	179
Figure 5-42: WPF Found by the Second Search Scheme with 19800 points of S_2 and Surrogate Models for the Aircraft Design Example	179
Figure 5-43: Comparison between the Two WPF's Found by the First Search Schemes with 9900 and 19800 Points of S_2 and Surrogate Models for the Aircraft Design Example	181
Figure 5-44: Comparison between the Two WPF's Found by the Second Search Schemes with 9900 and 19800 Points of S_2 and Surrogate Models for the Aircraft Design Example	181
Figure 5-45: Comparison between the Two WPF's Found by the Two Search Schemes with 19800 Points of S_2 and Surrogate Models for the Aircraft Design Example	182
Figure 5-46: Comparison between the Two WPF's Found by the Two Search Schemes with 19800 Points of S_2 and Original CA's for the Aircraft Design Example.....	183
Figure 5-47: Comparison between the Two WPF's Found by the First Search Schemes with 19800 Points of S_2 , Surrogate Models, and Original CA's for the Aircraft Design Example	183
Figure 5-48: Comparison between the Two WPF's Found by the Second Search Schemes with 19800 Points of S_2 , Surrogate Models, and Original CA's for the Aircraft Design Example	184
Figure 5-49: Example of the DSM of a Multidisciplinary RLV Design Problem	188

Figure 5-50: WPF Found by the First Search Scheme with 39600 points of S_2 and Surrogate Models for the RLV Design Example.....192

Figure 5-51: WPF Found by the Second Search Scheme with 39600 points of S_2 and Surrogate Models for the RLV Design Example193

Figure 5-52: Comparison between the Two WPF's Found by the Two Search Schemes with 39600 Points of S_2 and Surrogate Models for the RLV Design Example.....194

Figure 5-53: Comparison between the Two WPF's Found by the Two Search Schemes with 39600 Points of S_2 and Original CA's for the RLV Design Example195

Figure 5-54: Comparison between the Two WPF's Found by the First Search Schemes with 39600 Points of S_2 , Surrogate Models, and Original CA's for the RLV Design Example.....195

Figure 5-55: Comparison between the Two WPF's Found by the Second Search Schemes with 39600 Points of S_2 , Surrogate Models, and Original CA's for the RLV Design Example.....196

LIST OF SYMBOLS AND ABBREVIATIONS

Symbols

$\langle \cdot, \cdot \rangle$	Dot product
α_i	i^{th} required probability in an inequality probabilistic constraint
β_i	i^{th} required probability in an equality probabilistic constraint
ε	Error or tolerable deviation of the ε -insensitive loss functions
θ	Parameter(s) to be determined of a function family, or a general statistical parameter
ρ	Correlation coefficient
C	Regulation factor of Support Vector Regression
Cov	Covariance
$f(\cdot)$	a function of ‘ \cdot ’
$f_i(X)$	i^{th} objective function of X vector
$f_X(X)$	Marginal distribution function of X (a single random variable or a vector of rand variables)
$f(y X)$	Distribution function of y conditioned on X vector
$f(y, X)$	Joint distribution function of y and X vector
$F(\cdot)$	Objective function vector or a function of ‘ \cdot ’
$g(X)$	A surrogate model of X vector
$g_i(X)$	i^{th} inequality limit state function
$g(X, \theta)$	Function family of X vector with θ as the parameters

$h_i(X)$	i^{th} equality limit state function
$I\{\cdot\}$	The indicator function of ‘ \cdot ’
$k(X, X')$	Kernel function of X and X'
$L(Z, \theta)$	Loss function of Z and θ
P_f	Probability of failure
P_s	The set of s sample points
$P(\cdot)$	Probability of ‘ \cdot ’
$r(X)$	Regression function of X vector
R^2	Coefficient of determination
R_s	Structural risk function
$R(\theta)$	Risk function of θ
s	Sample size
S	A sample
sup	Supremum (the least upper bound)
th_i	i^{th} threshold in a deterministic constraint
U	A uniform random variable
x_i	i^{th} design variable (either random or deterministic)
X	Vector of design variables, or a random variable
\bar{X}	Normalized value of X
X_i	i^{th} sample point (value of X vector)
y	A true response

\bar{y}	Normalized value of y , or the mean value of a sample of variable Y
\hat{y}	A predicted response of a surrogate model
y_i	i^{th} response
z_i	i^{th} limit state function
Z	A vector of limit state functions
$\frac{\cdot}{\cdot}$	Normalized value

Abbreviations

\$/RPM	Average required yield per revenue passenger mile
AAO	All-at-Once method
AIAA	American Institute of Aeronautics and Astronautics
AIC	Akaike information criterion
AICC	Modified Akaike information criterion
ALCCA	Aircraft Life Cycle Cost Analysis
ANN	Artificial neural network
BIC	Bayesian information criterion
BICC	Modified Bayesian information criterion
BLISS	Bi-level integrated system synthesis method
BPDF	Bandte PDF method
CA	Contributing analysis
CDF	Cumulative distribution function
CO	Collaborative optimization method

CPDF	conditional probability density function
CV	Control variable
CX	Coupling variable between two disciplinary analyses
DoD	Department of Defense
DoE	Design of Experiment
DSM	Design structure matrix
EDF	Empirical distribution function
EDS	Extended design space
ERBF	Exponential radial basis function
ERM	Empirical risk minimization principle
FLOPS	Flight optimization system
FPI	Fast probability integration, or fixed point iteration method
GA	Genetic Algorithm
GP	Gaussian Process
GRBF	Gaussian radial basis function
HOT	Higher order term
HS	Hammersley sequence sampling
IC	Information criterion
ISP	Specific impulsive value
JPA	Joint probability assessment
IPDF	Inverse transformation method
JPDM	Joint probabilistic decision making
JPMOMDO	Joint probabilistic, multi-objective, multidisciplinary optimization

KG	Kriging
KKT	Karush-Kuhn-Tucker conditions
LCC	Life cycle cost
LHC	Latin hypercube sampling
LSF	Limit state function (the right hand side of a deterministic constraint)
MADM	Multi-attribute decision-making technique
MAE	Maximum absolute error
MARS	Multivariate adaptive regression splines
MC	Monte Carlo or Monte Carlo sampling
MCO	Modified collaborative optimization method
MCS	Monte Carlo simulation
MDDV	Multidimensional data visualization
MDL	Minimum description length
MDO	Multidisciplinary design optimization
MFE	Model fitting error
MFOSM	Multi-response first order second moment method
MOO	Multi-objective optimization
MPE	Model predicting error
MPP	Most probable point
MSE	Mean square error
NASA	National Aeronautics and Space Administration
NC	Normal Constraint method
NPDF	Nataf transformation method

NV	Noise variable
OA	Orthogonal array or orthogonal array sampling
OBD	Optimizer based decomposition method
OEC	Overall evaluation criterion method
PC	Probabilistic constraint
PDF	Probabilistic density function
PF	Pareto frontier
POS	Probability of success
RBF	Radial basis function
RCV	Random cross validation
RDS	Robust design simulation
RLV	Reusable launch vehicle
RMSE	Root mean square error
RSM	Response Surface Methodology
RSSVR	The hybrid surrogate model of second order RSM and SVR
SA	Simulated Annealing
SM	Surrogate model
SRM	Structural risk minimization principle
SVM	Support Vector Machine
SVR	Support Vector Regression
TIES	Technology identification, evaluation, and selection
TOPSIS	Technique for order preference by similarity to ideal solution
Trn	Training

UD	Uniform design
UUD	Unique uniform design
VC dimension	Vapnik-Chervonenkis dimension
WPF	Weak Pareto frontier

SUMMARY

One inherent impetus to the development of new or better design methods is the challenge of solving realistic design problems of complex systems, where ‘realistic’ means no simplifications have been made to the design problem except for the mathematical abstraction of the design problem itself. Specifically, realistic conceptual design problems of complex systems have four common features: multidisciplinary, multi-objective, design decisions being made in the presence of uncertainties, and decisions being made in a relatively short time period with limited resources. Those realistic conceptual design problems are either for design concept generation and selection or for design alternative generation and selection. Although design has been viewed as a discipline for more than three decades, the current state-of-the-art design methods have limitations and in many cases are not suitable to handle realistic conceptual design problems. This will be particularly true for the cases of design alternative generation and selection where revolutionary design concepts are considered. This drives the need for a new framework to solve realistic conceptual design problems of design alternative generation and selection. Considering the fact that high fidelity but time consuming tools are used to generate the training sample for surrogate model construction when a design is revolutionary, this new framework in turn requires a hybrid surrogate modeling method to achieve high accuracy for many kinds of problems with a small training sample. This new framework also requires a surrogate model selection advisor to choose the best surrogate model for a given complex physics-based model based on a balance between model accuracy and complexity.

The purpose of this research is to provide such a framework. The proposed framework combines separately developed multidisciplinary optimization, multi-objective optimization, and joint probability assessment methods together but in a decoupled way, to solve joint probabilistic constraint, multi-objective, multidisciplinary optimization problems that are representative of realistic conceptual design problems of design alternative generation and selection. The intent here is to find the Weak Pareto Frontier (WPF) solutions that include additional compromised solutions besides the ones identified by a conventional Pareto frontier. This framework starts with constructing fast and accurate surrogate models of different disciplinary analyses in order to reduce the computational time and expense to a manageable level so that the design space can be explored quickly, obtain trustworthy probabilities of the probabilistic constraints (PC) and WPF, and so as to enable conceptual design decision making in shorter time period.

A new hybrid method is formed that consists of the second order Response Surface Methodology (RSM) and the Support Vector Regression (SVR) method capturing the global tendency and the local nonlinear behavior respectively. The purpose of forming this hybrid method is to provide a method that can achieve high accuracy for many kinds of problems with a small training sample. The three parameters needed by SVR to be pre-specified are selected using practical methods and a modified information criterion that makes use of model fitting error, predicting error, and model complexity information. The model predicting error is estimated inexpensively with a new method called Random Cross Validation. In order to select a surrogate model without unnecessary complexity from RSM, SVR, and the hybrid method, this modified information criterion is also used as a surrogate model advisor to select the best surrogate model for a given problem.

A new neighborhood search method based on Monte Carlo simulation is proposed to find valid designs that satisfy the deterministic constraints and are consistent for the coupling variables featured in a multidisciplinary design problem, and at the same time decouple the three loops required by the multidisciplinary, multi-objective, and probabilistic features. Two schemes have been developed. One scheme finds the WPF by finding a large enough number of valid design solutions such that some WPF solutions are included in those valid solutions. Another scheme finds the WPF by directly finding the WPF of each consistent design zone that is made up of consistent design solutions. Then the probabilities of the PC's are estimated, and the WPF and corresponding design solutions are found.

Three pure mathematical model fitting problems are used to demonstrate that the hybrid method of RSM and SVR really can obtain more accurate surrogate models with better results where sometimes the (second order) RSM, SVR, and Neural Network methods can not fit a given problem well with a small training sample. This illustrates the need for the hybrid method.

Three two-objective and one three-objective deterministic optimization problems are used to demonstrate that this framework can find the true weak Pareto frontier. The results show this framework can be used for many types of problems, such as cases of multiple-to-one mapping from design solutions in the design space to objective points in the objective space, problems of which the WPF is made up of spatially disjointed segments, and problems with constraints and more than two objectives.

A typical aircraft design problem and a reusable launch vehicle design problem under probabilistic constraints are solved to demonstrate the feasibility of this framework

for engineering-based problems. The results of these two design problems show that both neighborhood search schemes can find the WPF. These results also show the methods to select the pre-specified parameters of SVR work well for engineering-based problems, the hybrid surrogate models are fast and accurate, and the surrogate model advisor can select the best surrogate model for a given problem or each response of an engineering-based problem.

1 INTRODUCTION AND MOTIVATION

Although design has been viewed as a discipline for more than three decades, the conceptual design methods already developed have limitations and many are unable at times to handle realistic conceptual design problems of complex systems for design alternative generation and selection, where ‘realistic’ means no simplifications have been made to the design problem except for the mathematical abstraction of the design problem itself. This drives the need for a new framework to solve such kind of problems, especially those involving revolutionary design concepts and technologies.

1.1 The Conceptual Design Process

Ever since the Industrial Revolution in England in the late 18th century, the extensive adoption of complex mechanisms such as automobiles, aircraft, and rockets has forced people to do design in a disciplined way. Before this, a design relied on a designer’s engineering intuition, talent, and experience, to compose concepts together to provide a viable solution for a given mission. Although design is still both a science and an art even in modern days [1] with engineering intuition and experience being necessary, it has undoubtedly become a discipline that provides the methodology to decompose mission or customer requirements, find and compose proper concepts in an organized or structured way, and finally provide a solution. Design as a discipline is based on mathematics, scientific and technical knowledge, other information of analytical disciplines, and information integration technology. Design as a discipline can provide better solutions such as the optimal solutions than engineering intuition or experience can, or even be the only means to provide viable solutions when new and complex

systems to be tackled are beyond the capability of an experienced designer. As complements to the discipline of design, however, engineering intuition and experience can provide good starting points or efficient details of the overall solution.

A formal definition of design was given by Blumrich in 1970 as “Design establishes and defines solutions to and pertinent structures for problems not solved before, or new solutions to problems which have previously been solved in a different way” [2]. This definition is the symbol of the discipline of design.

Engineering design can be divided into three major phases: conceptual design, preliminary design, and detailed design [3]. Conceptual design can be further divided into need identification and problem definition, concept generation, concept selection [1], and design alternative generation and selection. A design concept is an idea that represents a family of similar design alternatives and eventually is described by a parametric model with some design variables to predict or estimate the performance, quality, and cost of this family of design alternatives, while a design alternative is a specific design resulting from specific values of the design variables [4]. In this research, ‘design solution’ is often used as a more familiar term to most people for ‘design alternative’.

All possible design concepts form the concept space. A good way to explore the concept space is to form a morphological matrix that consists of functions and sub-functions required by a design problem and corresponding possible ‘hows’ [1], such as different technologies, number of engines, wing shapes, etc. A design concept is a combination of such ‘hows’. For example, design concept 1 is a combination of 3 engines of technology 1, straight wing, low wing, etc; and design concept 2 a combination of 2

engines of technology 2, swept wing, middle wing, etc. Therefore the concept space has a combinatorial property.

Each design concept has a design space, which is formed by the ranges of the design variables of the parametric model of this design concept. A design alternative is thus a point in a design space, or a sized design concept, as shown in Figure 1-1.

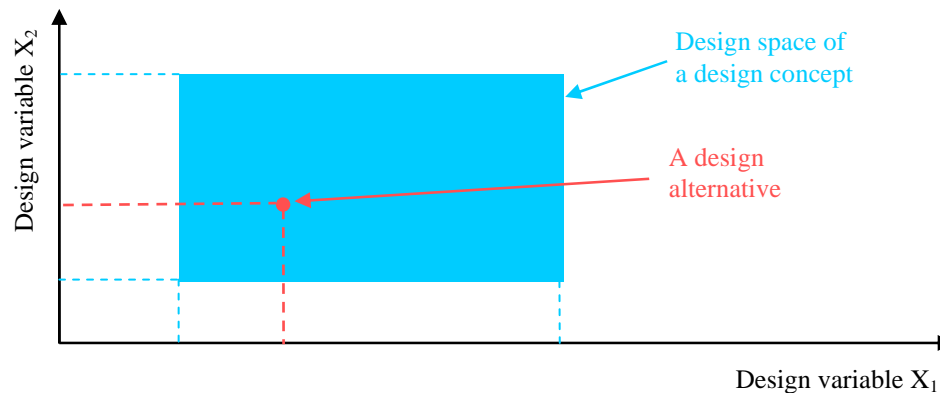


Figure 1-1: The Design Space and a Design Alternative of a Design Concept

The output of the conceptual design phase is a family of design alternatives. One example is the design of the Korean trainer T-50, as shown in Figure 1-2. During the conceptual design, 19 concepts were generated and evaluated. Some of the concepts are sized, and at the end three design alternatives are provided for preliminary design. Another example is the design of a notional wing spar, as shown in Figure 1-3. The design concept is selected as a flat plate. A design alternative is provided when the length and depth of this plate are determined according to wing span, airfoil shape, and location.

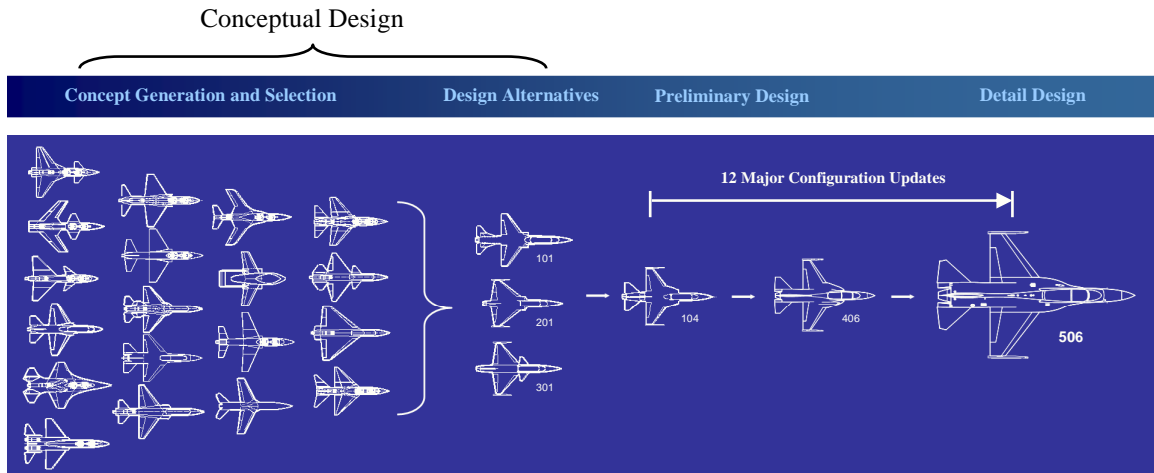


Figure 1-2: Design Process of Korean Trainer T-50 (Adapted from [5])

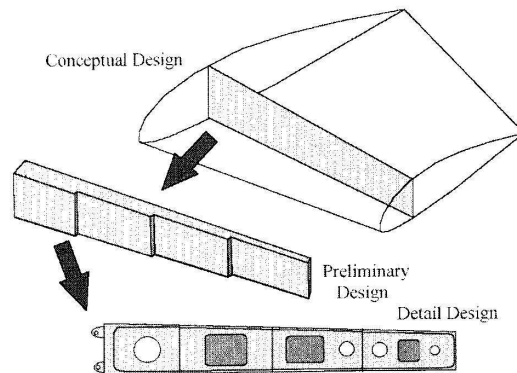


Figure 1-3: Design Process of a Wing Spar [3]

1.2 The Design Process Paradigm Shift and Required Design Methods

A design process paradigm shift is underway and proposes a change in the way complex systems are being designed. The new design process paradigm entails tradeoffs between conflicting objectives, integration of life-cycle disciplines, and a probabilistic design approach to handle uncertainties.

1.2.1 The Design Process Paradigm Shift and Its Implications

Today's design paradigm is experiencing a shift from design for performance, without much consideration of the cost implications, to design for affordability and

quality with a life-cycle emphasis. The goal of this new paradigm is to design complex systems with high quality at a competitive cost while accounting for the life-cycle behaviors of those systems [6]. Affordability as a concept is also introduced, under this paradigm as the ratio of system effectiveness to system cost, or in other words, the balance between performance (or more generally benefit) and cost. The life cycle of a product consists of 15 processes or stages if one looks closely, and can be divided into the pre-market and market phases [1]. For aircraft, the life cycle can be simplified as 6 main stages: conceptual design, preliminary design, detailed design, production, service, and retirement.

In the traditional design approach, the design goal is to maximize performance without much consideration of the cost. Usually a handful of design concepts or design alternatives are selected for further analysis after the conceptual design phase and usually one design alternative is selected for further development. Thus there is a limited amount of design freedom in this process, where design freedom is the ability to generate viable engineering alternatives and make design changes during the product development stages before product release [7]. Consequently, the design decisions made in the early stages such as conceptual and preliminary designs determine a large portion of the total life cycle cost (LCC) committed, and with few exceptions high LCC is incurred. Those decisions also can have significant impact on the whole life cycle of a product, including product quality and customer satisfaction.

This situation can be changed in the new design paradigm by shifting to a more gradually decreasing design freedom curve and a more gradually increasing cost committed curve. To achieve the two new curves, downstream design knowledge must be

brought to the early design phases such as conceptual design in order to make educated decisions (increasing knowledge), where design knowledge is the information about the product, the process and operational environment; performance-cost tradeoffs must be made in early design phases; the design freedom must be kept open by adopting **probabilistic** design approach to provide a family of design alternatives in order to mitigate the effects of uncertainties [7]; and higher fidelity knowledge should be brought into the conceptual design phase through higher fidelity design and analysis tools [6]. With more design freedom and higher fidelity knowledge, the cost committed by the design decisions will be decreased. Thus the total life cycle cost will be decreased. This “cost-knowledge-freedom” interaction from conceptual design to production is shown in Figure 1-4, where the cost is the LCC.

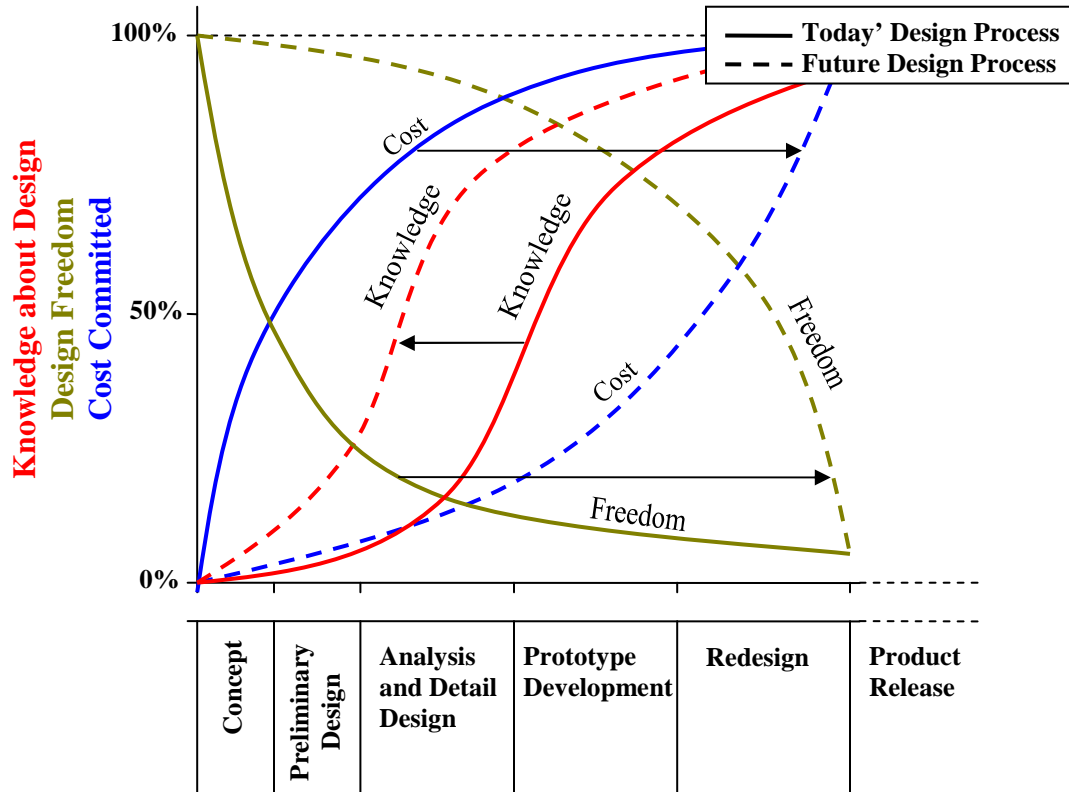


Figure 1-4: "Cost-Knowledge-Freedom" Interaction and Shift for Future Design Paradigm [6, 7]

The above design paradigm shift requires handling more conflicting design objectives. Traditional design has only performance objectives; modern design has to handle conflicting performance and cost objectives. With conflicting objectives, there is no best solution, there are only best compromised solutions or efficient solutions, typically called Pareto frontier (PF) solutions. As shown in Figure 1-5, there are many possible solutions in the objective space, and the Pareto frontier solutions are at the edge of the cloud of points. Therefore, Pareto frontier solutions enable efficient tradeoffs between performance and cost, or selection by the identification of the User. Pareto frontier solutions are a locus of different solutions; with these solutions, the design freedom is kept open.

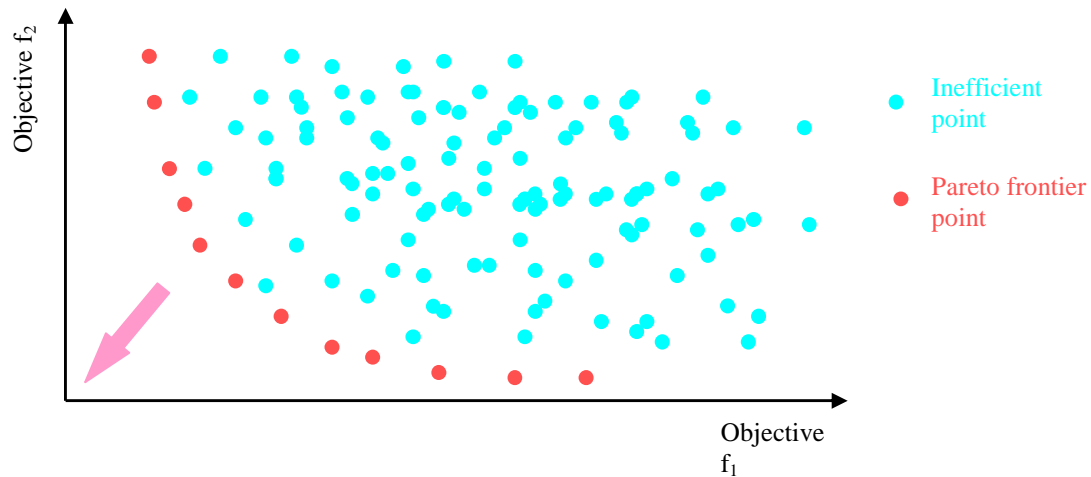


Figure 1-5: Pareto Frontier Points and Inefficient Points in the 2D Objective Space

The above design paradigm shift also requires handling more disciplines involved in the life cycle and more design requirements or constraints during the conceptual design phase. In addition to conventional disciplines such as propulsion, performance, and structure, other downstream disciplines of the life cycle should be included, such as

stability and control, economics (for cost and profit), manufacturing, and safety, as shown in Figure 1-6. Now requirements or constraints from different levels must be considered at the same time, including traditional customer requirements and technical standards, and new regulatory requirements such as emission and noise. For aircraft design, relevant different levels may include discipline, vehicle system, transportation system, and global system levels. Here transportation system includes aircraft, airport, flight route facilities, etc; and global system includes the country and the Earth in the viewpoints of national economics and ecological impacts.

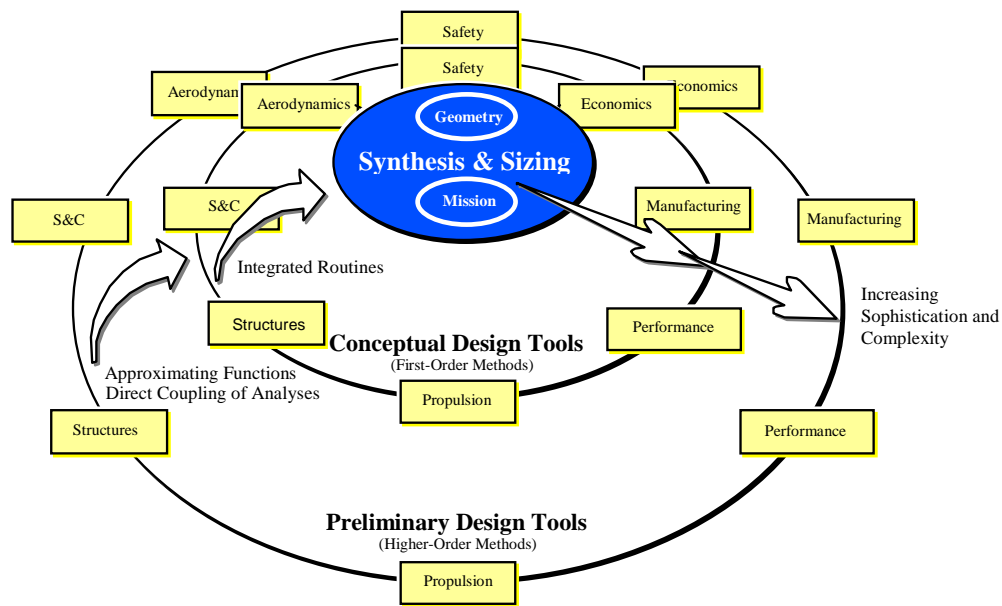


Figure 1-6: Variable Fidelity of Aircraft Synthesis and Sizing [7]

With respect to the conceptual design methods, this design paradigm shift implies the following capabilities or features are desired: first, to handle multiple design objectives and find Pareto frontier solutions for efficient tradeoffs between cost and performance; second, to include life-cycle disciplines and to handle the interactions among those disciplines; third, to perform probabilistic design in order to handle the

uncertainties and keep the design freedom open; last, to perform rapid assessments, since the conceptual design phase is relatively very short compared with the total life cycle and there is an enormous number of possible concepts and design alternatives that need to be investigated.

With respect to the mathematical model of a conceptual design problem, the design paradigm shift implies a realistic model is desired. Here ‘realistic’ means no simplifications have been made to the design problem except for the mathematical abstraction of the design problem itself. The common features of the realistic conceptual design problems include: multiple conflicting design objectives; multiple disciplinary analyses with coupling variables; probabilistic constraints to capture the effects of uncertainties; and use of accurate or high fidelity disciplinary knowledge due to profound effects of design decisions on LCC.

1.2.2 Concepts of Multi-Objective Optimization Methods

As mentioned previously, the design of complex systems needs to handle conflicting objectives and Pareto frontier solutions enable efficient tradeoffs among those conflicting objectives. The reason for Pareto frontier solutions is discussed in more detail here.

In most cases, the quality of the complex systems must ultimately be assessed by more than one criterion, and all of the corresponding objectives of those criteria should be optimized simultaneously. Often, the objectives are conflicting in such a way that optimization of a single objective leads to poor performance for other objectives. Generally speaking, there are many potential design solutions to a multi-objective optimization problem (MOO). During the early stages of decision-making, the designer

or technical engineer often has little information about the relative importance of the individual objectives, or which criterion is more important. These decisions are usually not made by the designer; instead, these decisions are preferences that vary depending on a stakeholder's viewpoint [8]. In the absence of a decision maker, all the objectives may be treated with equal importance, and a whole family of such solutions of which no objective can be further improved without degrading any other objectives must be found. These solutions are denoted as Pareto frontier solutions.

From the above description, there are no other solutions better than the PF ones in terms of all objectives. Therefore, if anyone of the PF solutions is selected according to a given criterion for tradeoff among the conflicting objectives, this PF solution is the best. In this sense, PF solutions enable efficient tradeoff since no efforts will be wasted on the inferior solutions.

Because of the high efficiency of PF solutions for tradeoff, the purpose of mathematical optimization is to give a variety of PF solutions, or in the ideal case to determine the entire set of PF solutions, to the User or customer who is the decision-maker. With the PF solutions, the User can determine the optimum design according to certain preferences for the objectives during the decision-making process. This is an important ability for engineering design, as shown in Ref. [9], "Pareto front techniques help define the biggest bang-for-buck so that, for instance, the DoD can decide on how much performance it can afford".

The user may also choose to relax the requirements so as to accept such design solutions for which at least one, but not all, objectives are better than those of other design solutions. This is a means by which the design solution space is opened up and

more trade-off choices become available. These relaxed MOO solutions are denoted as weak Pareto frontier (WPF) solutions.

1.2.3 Concepts of Multidisciplinary Optimization Methods

Multidisciplinary design optimization methods are formulated to integrate different disciplinary analyses together to handle the interactions and enable a concurrent engineering process to solve such design and optimization problems efficiently.

The different disciplines interact with each other through the coupling variables among those disciplines. A coupling variable (denoted as CX) is both an output of a disciplinary analysis and an input of another where the first disciplinary analysis directly or indirectly needs input information from the second one. A single-discipline design such as the design of the engine in the aircraft design may have coupling variables if some outputs are fed back.

Coupling variables complicate multidisciplinary design. There are three main impacts of these variables. First, design freedom is reduced since only some design points or solutions in the system level design space lead to converged values for the coupling variables themselves. These points are called consistent design points, and form disjointed zones in the design space, which are called consistent design zones. Second, those variables require many iterations of the multidisciplinary analysis in order to find every single consistent design point since these consistent design zones are disjointed. In other words, equality constraints entailed by those variables in the multidisciplinary analysis process complicate the design problem. Third, special solving procedures are required to decouple the complex interactions introduced by coupling variables to find consistent design points.

If there are (explicit) design constraints in the multidisciplinary design, only some of the consistent design points can satisfy those constraints. These points are called feasible design points, or herein, candidate design points. Consequently, the feasible design space consists of disjointed feasible design zones that are inside of and no larger than the corresponding consistent design zones. Obviously more iterations of multidisciplinary analysis are required to find every feasible design point from the disjointed consistent design zones. Sometimes a consistent zone may contain disjointed feasible design zones, and this makes the design even more complicated.

1.2.4 Concepts of Probabilistic Design Methods

The conceptual design of complex systems is probabilistic in nature, such that decisions are made in the presence of uncertainties. Simply speaking, uncertainty is the incompleteness of design knowledge, or a difference between reality and what is expected [7]. In more detail, uncertainties are caused by ambiguity of the requirements, variations in material properties, incomplete knowledge of the manufacturing process and operational environment such as variations in manufacturing precision and loading conditions, modeling assumptions, and other sources [6].

Uncertainties can significantly affect the decision-making process. Traditional multidisciplinary design optimization methods use a deterministic approach so that the optimal design solutions are frequently pushed to the limits of design constraint boundaries, leaving little or no room to accommodate uncertainties in system input, modeling, simulation, and operation environment [10]. As a result, those design solutions may be highly sensitive to the variation of the uncertain factors. This can lead to serious performance loss suffering from the high likelihood of undesired events such as some

extreme off-design conditions, or being conservative and consequently not economically viable.

Most often the effects of uncertainties are embodied in the probabilistic constraints (PC) [11]. A probabilistic constraint is that the probability of satisfying one constraint must be greater than a prescribed level. This constraint is called limit state function (LSF). For example, to consider the variation in material properties and/or load conditions, the PC can be stated as the probability that the maximum stress in a structure is less than a given level must be greater than 99.9%, or that the failure probability must be less than 0.1%. To consider system-level requirements subject to future changes, the designer can use a stricter-than-current requirement, for example, if the takeoff distance is to be less than 5,500 feet, a stricter requirement can be made as that the takeoff distance must be less than 5,000 feet; and the PC can be stated as that this stricter requirement must be satisfied with a probability greater than 85%.

To consider the effects of uncertainties, the conceptual design of complex systems has to adopt a probabilistic design approach. The probabilistic design approach is quite different from the (traditional) deterministic design approach. As shown in Figure 1-7, an imaginary constraint analysis for average required yield per revenue passenger mile (\$/RPM) and cruise speed, the deterministic constraints are fixed, defined by only two lines, while the probabilistic constraints are not fixed or well defined, instead, these probabilistic constraints are represented by two bands. Consequently, use of the deterministic design approach provides one design alternative, while the probabilistic design approach yields a family of design alternatives.

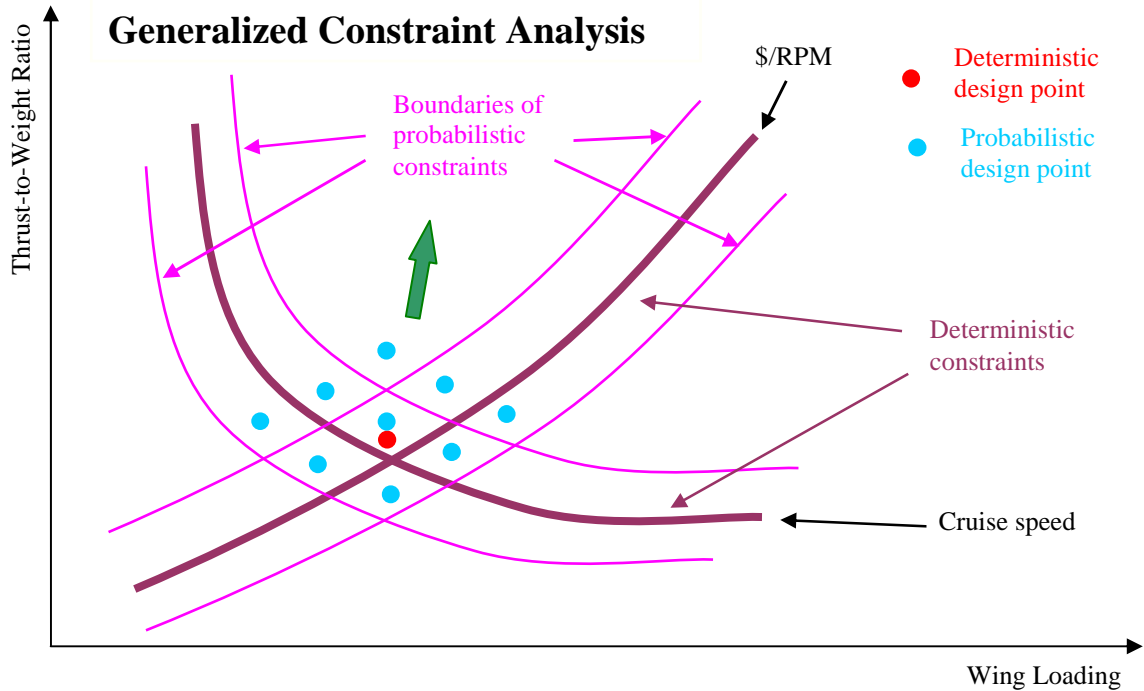


Figure 1-7: Constraints of Deterministic and Probabilistic Design Approaches

For evolutionary designs, there are a small number of uncertainties to be considered, and variations of those uncertainties are usually small; therefore, only a few design alternatives are needed. For revolutionary designs, a lot more assumptions are being made, and therefore there are more uncertainties, and variations are greater; therefore, a much larger set of design alternatives must be considered for examination.

As mentioned previously, the conceptual design phase includes design concept generation and selection, and design alternative generation and selection. The probabilistic design processes for design concepts and design alternatives are different. This is due to two main differences. First, the distributions are different. In probabilistic design, the designer actually controls only the nominal values of the design variables, such as the mean values. The distributions of a design concept are for those nominal values. Usually, a uniform distribution is used. The distributions of a design alternative

are for the values of the design variables about a set of nominal values. Usually these distributions are not uniform. This difference is illustrated in Figure 1-8.

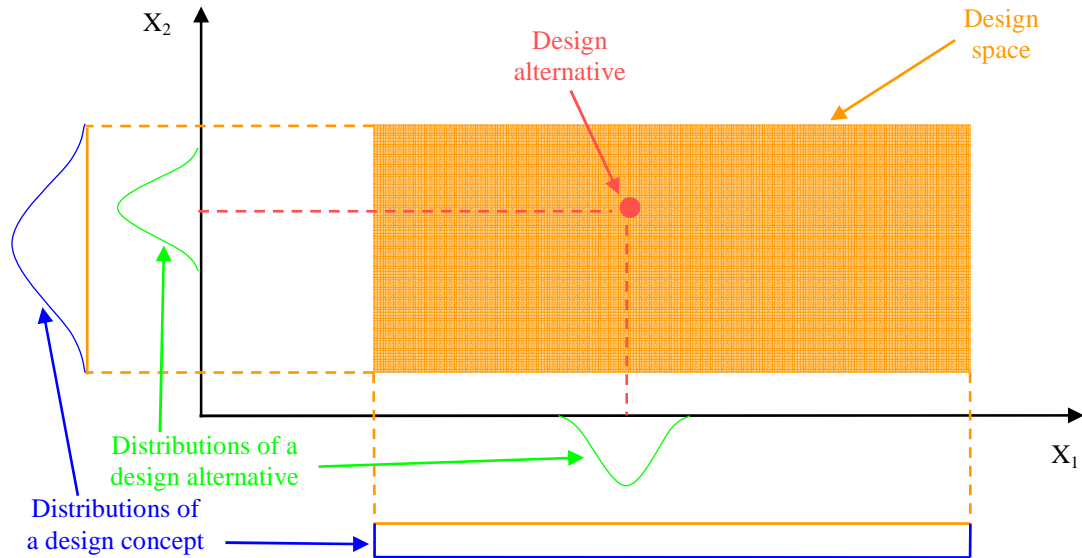


Figure 1-8: Distributions of a Design Concept and a Design Alternative in Probabilistic Design

Second, the constraints imposed on the design alternatives are different. When assessing the probability of a design concept via sampling techniques for design concept generation and selection, each sampling point is checked to see if all deterministic constraints are satisfied. Here a sampling point is actually a design alternative. Therefore, only deterministic constraints are imposed on each design alternative of this concept, although probabilistic constraints are imposed on this concept. When assessing the probability of a design alternative via sampling techniques for design alternative generation and selection, similarly each sampling point is checked to see if all deterministic constraints are satisfied. What is different in this case is that a sampling point is actually a possible physical realization of this design alternative. Therefore, in this case probabilistic constraints are imposed on a design alternative, although

deterministic constraints are imposed on each possible physical realization of this design alternative.

In most cases, the PC's are not independent because those constraints share some variables, either random or deterministic. Because of those shared variables, the effects of uncertainties propagate around, and thus those PC's must be satisfied jointly. Obviously, the effect of the PC's is that under PC's some of the WPF solutions found by traditional methods under **deterministic** constraints are not eligible and have to be discarded.

1.3 The Need for a New Framework for Design Alternative Generation and Selection

Many conceptual design methods have been developed to accommodate the new design paradigm. These methods follow the Integrated Product and Process Development (IPPD) methodology [7, 12] that systematically integrates and applies all life cycle disciplines into the early design phases. Examples of such methods are the Technology Identification, Evaluation, and Selection (TIES) method [6], Joint Probabilistic Decision Making (JPDM) method [8], and Robust Design Simulation (RDS) method [13], to name a few. Those methods are good for realistic problems of design **concept** generation and selection, and have been successfully applied in many design projects.

However, those methods are not suitable for realistic problems of design **alternative** generation and selection because of some of the following limitations. Some of those methods are used with monolithic legacy codes that are suited for evolutionary designs, but not suitable for revolutionary designs that are out of the scope of those codes. Usually there are no explicit treatments for the coupling variables. When solving a revolutionary design, this is a problem because separate disciplinary analyses are used in

this case and the convergency of coupling variables has to be treated by the designer, instead of the disciplinary analyses. Some of those methods treat multiple objectives with a single Overall Evaluation Criterion (OEC). The problem of this treatment is that the solutions are not guaranteed true Pareto frontier solutions [14]. The last limitation is that usually there is no algorithm to search for a design alternative that satisfies all probabilistic constraints. This is because usually sample filtering type methods (e.g. Monte Carlo sampling and filtering) are used to handle the combinatorial property of the concept space and to assess the design concept probabilities by sampling design alternatives. When selecting design alternatives, these types of methods are inefficient and can not guarantee PF solutions since it is possible that none of the sampling points is a PF solution, or in the worse case none satisfies all probabilistic constraints.

Therefore, there is no suitable method for realistic conceptual design problems of design alternative generation and selection according to the previous investigation in terms of treatment of probabilistic constraints, explicit search for PF, and explicit treatment of coupling variables. This is a gap that needs to be filled. And to fill this gap is the main motivation of this research.

It is important to form a new method or framework for realistic conceptual design problems of design alternative generation and selection. Because of the limitations, current conceptual design methods can only be used to solve simplified problems of design alternative generation and selection. For example, when the design concept is revolutionary, the step to check the convergency of coupling variables may be skipped because of lack of explicit treatment for coupling variables; or a design alternative may be accepted without checking if it is a PF solution because of lack of an algorithm to

search for design alternatives that satisfy all probabilistic constraints. The problem with design alternatives obtained from simplified problems is that the solutions may lead to increased risk and cost in the later design phases. There may be more and serious risk and cost for revolutionary designs because of more uncertainties and greater variation.

1.4 Research Objective

In a general sense, AIAA MDO Technical Committee [9] and NASA Langley Research Center [15] summarized the elements needed by new design methods in the aerospace industry, which include techniques to handle Pareto Frontiers with multiple design objectives, loosely coupled multidisciplinary design optimization (MDO) frameworks or architectures to efficiently handle a wide variety of problems, better approximation methods than current popular methods such as RSM and ANN to reduce computing time and cost, and algorithms to account for uncertainties and perform optimization under uncertainties at conceptual through detailed design phases.

Obviously, the above general expectations for new methods in the aerospace industry are a detailed version of the desired features of a conceptual design method in order to accommodate the new design paradigm. Considering the above general expectations, the primary goal of this research is to formulate a practical framework to solve realistic conceptual design problems for design alternative generation and selection, where ‘practical’ means having the desired elements, including handling multiple design objectives and finding weak Pareto frontier (WPF) solutions; handling multiple life-cycle disciplines and the interactions among these disciplines in a loosely coupled way; performing probabilistic design to account for uncertainties; and performing rapid

assessment and enabling use of accurate or high fidelity knowledge by accurate approximation methods.

One note is that this framework should find weak Pareto frontier solutions instead of PF solutions since WPF can provide additional compromised design solutions for tradeoff than PF as discussed previously. And later on, a short term of ‘realistic conceptual design problem’ will be used for ‘realistic conceptual design problem of design alternative generation and selection’.

The above expectations for the new framework require the implementation of joint probabilistic (constraint), multi-objective, multidisciplinary optimization (JPMOMDO) and finding the WPF solutions. This framework is thus called “a framework for the determination of weak Pareto frontier solutions under probabilistic constraints”.

2 BACKGROUND

In this chapter, the necessary background literature is reviewed according to the desired elements for the new framework. First of all is the mathematical statement of the problem that represents a realistic conceptual design problem (of design alternative generation and selection) that the new framework should solve. Since the new framework should be based on accurate surrogate models, a literature review of the state-of-the-art in DoE and surrogate-modeling methods, and related surrogate-modeling concepts is presented. Additional methods relevant to this framework are investigated and include: model assessment and model selection methods, joint probability assessment methods, probabilistic design methods, multi-objective optimization methods, and multidisciplinary optimization methods.

2.1 Mathematical Statement of Realistic Conceptual Design Problems and Solving Considerations

A joint probabilistic (constraint), multi-objective, multidisciplinary optimization problem, which represents a realistic conceptual design problem to be solved by the new framework, can be represented by the following mathematical model:

$$\text{Minimize: } F(X) = [f_1(X), f_2(X), \dots, f_e(X)]^T \quad \text{objective functions} \quad (2.1)$$

Subject to:

$$P(g_j(X) \leq th_j) \geq \alpha_j \quad j = 1, m \quad \text{inequality PC's} \quad (2.2)$$

$$P(h_k(X) = th_k) \geq \beta_k \quad k = 1, l \quad \text{equality PC's} \quad (2.3)$$

$$CX_q^I = CX_q^O \quad q = 1, c \quad \text{coupling variable constraints} \quad (2.4)$$

$$X_i^l \leq X_i \leq X_i^u \quad i = 1, n \quad \text{side constraints} \quad (2.5)$$

where

$$X = [x_1, x_2, \dots, x_n]^T \quad \text{(random) design variables}$$

$$y_1 = f_1(X), y_2 = f_2(X), \dots, y_e = f_e(X) \quad \text{responses/objectives}$$

$$CX_q^I, CX_q^O \quad \text{input and output coupling variables}$$

$$g_j(X), h_k(X) \quad \text{limit state functions}$$

$$th_j, th_k \quad \text{thresholds}$$

$$\alpha_j, \beta_k \quad \text{required probabilities}$$

The constraints $g_j(X) \leq th_j$ and $h_k(X) = th_k$ are deterministic constraints. The functions in Equations 2.1, 2.2, and 2.3 may be linear or non-linear functions. Those constraints may be explicit or implicit functions of design variables X , and may be evaluated by analytical or numerical techniques depending on if those constraints are explicit functions. For most engineering problems, constraints are non-linear and implicit functions and have to be evaluated numerically using complex evaluation techniques such as finite element method.

The values of the design variables X are actually nominal values (such as mean values and the most probable values), and for each value there is a probabilistic distribution. Thus, the objective of this optimization problem is to find the set of nominal values of the design variables that satisfies the probabilistic constraints as well as minimizes the objective functions by adjusting the nominal values of the design variables.

With almost no exception, this JPMOMDO problem has to be solved iteratively, even with explicit functions. Therefore this iterative solving process requires a large number of complete analyses or simulations. When Equations 2.1, 2.2, and 2.3 are explicit functions, evaluations can be done very fast and the time needed to obtain the final optimal result is not a concern; however, for most engineering problems, the time needed to obtain the final optimal result is enormous and thus a big concern.

First, these equations are not explicit to the designer because the computer models are complex and often only executable binary files are available. Therefore it is not trivial to perform sensitivity analyses or calculate gradients (derivatives) that are used to accelerate the optimization process, whereas it is almost at no cost with explicit functions. Second, a single complete analysis or simulation needs a non-trivial amount of time because of the complex evaluation techniques adopted. Third, the multimodal nature of engineering problems often force the designers to use non-gradient based optimization techniques such as Genetic Algorithms (GA) or Simulated Annealing (SA) in order to avoid being trapped at a local minimum and to find the global minimal solution, but require a huge number of complete analyses or simulations. Last, a multi-objective optimization problem needs much more complete analyses or simulations than single-objective optimizations.

Therefore, it is impractical for real engineering problems to use exclusively complex analysis or simulation models for the purpose of optimization. A preferable strategy is to use surrogate models of complex high-fidelity models to reduce time consumption during optimization [16]. Actually, almost all multidisciplinary design optimization methods, such as Bi-level Integrated System Synthesis method (BLISS) [17]

and Collaborative Optimization method (CO) [18], to name a few, strongly recommend use of surrogate models to ensure good performance or efficiency in terms of time consumption. Depending on the formation of the mathematical representation of the design, the objectives and/or the constraints may be approximated with surrogate models.

Additionally, to solve the JPMOMDO problem, a loosely coupled or completely decoupled architecture is needed. If no special means are taken, a JPMOMDO problem can be solved in a nesting-loop approach, as shown in Figure 2-1, and three nesting loops are entailed. In this way, the computational time and load would be again too great to be accepted even when surrogate models are used. For example, if each loop requires 1,000 iterations, those three nesting loops require 1,000,000,000 iterations by product. Therefore, the solving process of the JPMOMDO problem must be loosely coupled or completely decoupled of the MDO, JPA, and MOO methods.

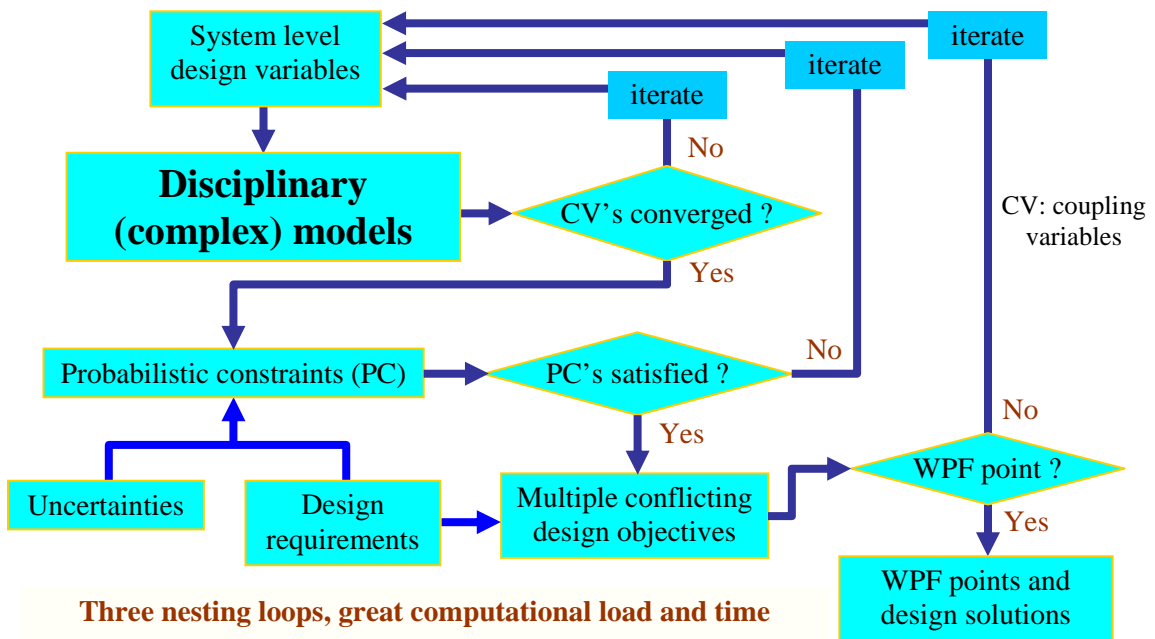


Figure 2-1: Potential Process of a Realistic Conceptual Design in a Nesting Loop Approach

In summary, the iterative solving process of the optimization methods requires a large number of analyses or simulations, thus surrogate models of the objectives and/or constraints are used to reduce the amount of time of the optimization process to an acceptable or manageable level; and even with surrogate models, in order to solve the JPMOMDO problem, a loosely coupled or completely decoupled architecture is needed.

2.2 Sampling Methods / Design of Experiments

Sampling methods or designs of experiment (DoE's) provide guidance for the selection of points to be evaluated such that the maximum information can be extracted from a minimum number of experiments. The classical DoE methods are briefly reviewed and some important modern sampling methods are summarized. In this section, three modern DoE methods are described. The overview of generic sampling methods, the overview of classical DoE methods, and other two modern DoE methods are provided in APPENDIX A.

2.2.1 Three Modern DoE's

With the development of science and technology, many physical systems or phenomena are studied so thoroughly that people describe those systems or phenomena by mathematical equations and perform simulation by solving these equations either analytically or numerically. When those equations and the methods are executed on a computer, it is called a computer experiment. A computer experiment is quite different from a corresponding physical experiment mainly from two aspects [19]. First, the result of a computer experiment is deterministic for a specific set of values of free variables including the design variables, whereas there are random errors for a physical experiment. Therefore the considerations of classical DoE's to minimize the effects of random errors

of the physical experiments, such as putting sample points at or near to the boundaries of the design space and replication of sample points, are not necessary or useful for a computer experiment. Second, it is very easy to change the levels of design variables in a computer experiment by just setting different numbers for the design variables, whereas for a physical experiment this may require making more prototypes or more elaborate and tedious work setting up the experimental conditions. Therefore, a computer experiment can have many more levels for the design variables than a physical one.

Due to the differences between a computer experiment and a corresponding physical one, the DoE's for computer experiments call for different considerations. The DoE's specifically developed for computer experiments are called modern DoE's. A consensus among researchers is that the sample points should be distributed throughout the design space, i.e. space filling, for computer experiments [20-22]. Modern DoE's are widely applied to computer experiments or simulations to construct surrogate models, and have been found to be able to provide a more accurate surrogate model than the classical DoE's [23]. Besides, Modern DoE's can improve the interpolation based surrogate-modeling methods [19], and minimize the bias error, which are caused by "the difference between the functional form of the true response trend, and the functional form of the assumed or estimated trend" [20]. Three popular modern DoE methods, i.e. LHC, HS, and MC are introduced in the following sections, and two more, i.e. OA and UD are introduced in APPENDIX A.

2.2.1.1 Latin Hypercube Sampling

Latin hypercube sampling is the first modern DoE developed specifically for computer experiments [24, 25]. The most popular algorithm for LHC sampling is:

$$x_i^j = \frac{\pi_i^j + U_i^j}{s}, \quad 1 \leq i \leq n, 1 \leq j \leq s \quad (2.6)$$

where x_i is the i^{th} design variable that is normalized to $[0,1]$ from its original interval $[x_i^l, x_i^u]$, n is the number of design variables, s is the number of sample points, π_i^1, \dots, π_i^s is an independent random permutation of the sequence of integers $0, 1, \dots, s-1$, and U is a uniform random value on $[0,1]$. The superscript j denotes the sample point number. There are $s!$ permutations of integers in π , all of which are equally likely to be picked without replacement. The interval of each design variable is divided into s subintervals, or “bins”, and the whole design space is divided into s^n bins.

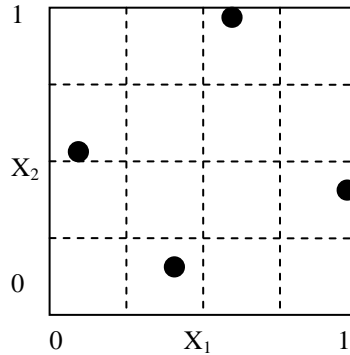


Figure 2-2: Example of Two Dimensional Latin Hypercube Sampling

A two dimensional example is shown in Figure 2-2, where $n = 2$ and $s = 4$. This design is generated using $\pi_1^1, \pi_1^2, \pi_1^3, \pi_1^4 = 0, 1, 2, 3$, and $\pi_2^1, \pi_2^2, \pi_2^3, \pi_2^4 = 2, 0, 3, 1$. These two π sequences are put in two consecutive columns in a matrix and each row of this matrix gives the (row, column) bin location of each sample points, i.e. (0,2), (1,0), (2,3) and (3,1), with bin (0,0) being at the lower left corner of Figure 2-2. The values of the two U sequences are approximately given as $U_1^1, U_1^2, U_1^3, U_1^4 = 0.37, 0.67, 0.35, 0.87$, and

$U_2^1, U_2^2, U_2^3, U_2^4 = 0.12, 0.67, 0.87, 0.68$. Using a similar method as for the π sequences, the values of the two U sequences gives the location of each sample point within its respective bin.

The Equation 2.6 gives only the algorithm for design variables with uniform distributions, but the LHC can be used for design variables with non-uniform distributions, as described in Ref. [26]. However, because of the deterministic computer experiments, the distributions of the design variables will not affect the accuracy of the surrogate model as long as the sample points are properly selected. Therefore, Equation 2.6 could be used for surrogate-modeling regardless of the real distributions of the design variables.

The LHC sampling has a significant advantage: the user can freely decide the number of sample points without restrictions to sample sizes that are specific multiples or powers of n . Besides, it can obtain good uniformity for small sized sample data. It has one main disadvantage, i.e. the freedom in the U sequence can cause large correlations among the design variables that may reduce the predicting accuracy of the surrogate model [23]. The correlation can be reduced to a user-specified level with some computation cost.

A derivative of LHC sampling, lattice sampling, is obtained by replacing the U sequence with a fixed value of 0.5, see Equation 2.7. The result is that each sample point is placed at the center of its respective bin, rather than randomly within the bin.

$$x_i^j = \frac{\pi_i^j + 0.5}{k}, \quad 1 \leq i \leq n, 1 \leq j \leq s \quad (2.7)$$

2.2.1.2 Hammersley Sequence Sampling

Hammersley sequence sampling is an alternative space filling sampling method [27]. Unlike LHC, Hammersley sequence sampling does not directly use a random number generator to generate the sampling points, but makes use of the randomness inside a prime number sequence. Its algorithm is as follows.

First, let's introduce the Radix-R notation of an integer. For a specific base R (e.g. 10) integer, p , can be represented in Radix-R notation as

$$\begin{aligned} p &= P_m P_{m-1} \cdots P_2 P_1 P_0 \\ &= p_0 + p_1 R + p_2 R^2 + \cdots + p_m R^m \end{aligned} \quad (2.8)$$

where $m = \lceil \log_R p \rceil$, and the square brackets, $\lceil \cdot \rceil$, denotes the integer portion of the number inside the brackets.

The inverse radix number function generates a unique number on the interval [0,1] by reversing the order of the digits of p around the decimal point, i.e.

$$\begin{aligned} \phi_R(p) &= \cdot P_0 P_1 P_2 \cdots P_{m-1} P_m \\ &= p_0 R^{-1} + p_1 R^{-2} + p_2 R^{-3} + \cdots + p_m R^{-m-1} \end{aligned} \quad (2.9)$$

Then the HS sample points are generated as

$$x_i^j(q^j) = \left(\frac{q^j}{s}, \phi_{R_1}(q^j), \phi_{R_2}(q^j), \dots, \phi_{R_{n-1}}(q^j) \right), \quad 1 \leq i \leq n, 1 \leq j \leq s \quad (2.10)$$

where x_i is the i^{th} design variable that is normalized to [0,1] from its original interval $[x_i^l, x_i^u]$, n is the number of design variables, s is the number of sample points, $q^j = j-1$, and R_1, R_2, \dots, R_{n-1} are any $n-1$ consecutive numbers of the prime number sequence (2,3,5,7,11,13,17,...).

A two dimensional example is shown in Figure 2-3, where $n = 2$ and $s = 20$.

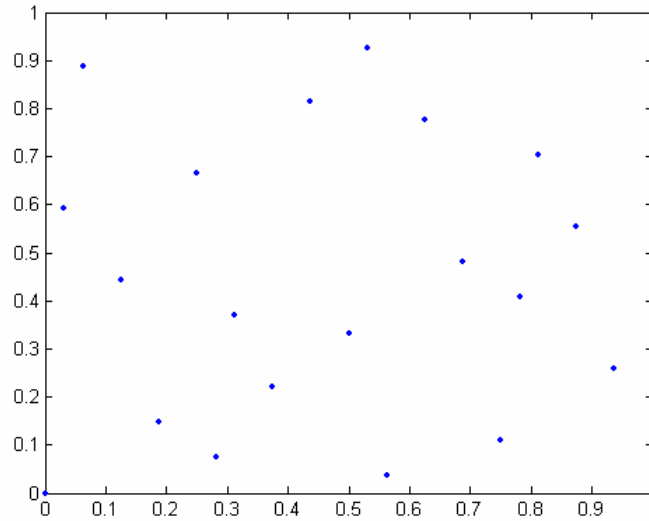


Figure 2-3: Example of Two Dimensional Hammersley Sequence Sampling

The HS sampling has two significant advantages. First, the user can freely decide the number of sample points. Second, the correlations among the design variables are very low, and this helps generate surrogate models with better predicting accuracy. Simpson et al [28] show that the HS sampling method tends to yield more accurate surrogate models in terms of lower model fitting errors. Therefore, the HS sampling method is a potential candidate for surrogate-modeling in this research. However, HS sampling has two main disadvantages. First, the distribution of the design variables can not be used to generate sample points. Second, if the sample size is small, the uniformity of the distribution of sample points is bad. As a rule of thumb, the sample size should not be less than $10n$, where n is the number of variables.

2.2.1.3 Monte Carlo Sampling

The (univariate) Monte Carlo sampling is exactly the pseudo-Monte Carlo sampling, of which “pseudo” implicates the use of a pseudo-random number generation algorithm that is intended to mimic a truly random natural process. It was first applied to computer

experiments in 1949 [29]. The MC sampling method is a genuine random sampling method if there is a true random number generator, and its algorithm makes use of the concept of the inverse transform method [30, 31].

Suppose $F(x)$ is the CDF of a random variable X , then the MC sampling method generates the sampling points as

$$x = F^{-1}(U) \tag{2.11}$$

where $F^{-1}(\cdot)$ is the inverse function of $F(x)$, and U is a uniform random variable of which values are generated by a (pseudo-) random number generator in computer experiments. Figure 2-4 shows the process to generate a sample point by MC sampling.

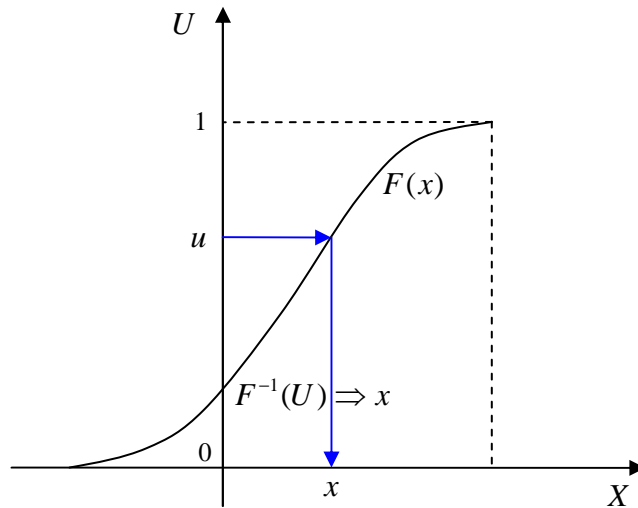


Figure 2-4: Univariate Monte Carlo Sampling Process

Although MC sampling is a genuine random sampling method, the randomness of its sampling process often leads to over- and under-sampled regions of the design space especially when the sample size is small. Therefore it should not be relied on unless a large sample size is used [31].

2.3 Surrogate-Modeling Methods

In this section, some basic concepts of surrogate-modeling methods and one popular surrogate-modeling method for engineering problems, i.e. the Response Surface Methodology, is reviewed, and the newly developing Support Vector Regression method is introduced. APPENDIX B describes three other popular surrogate-modeling methods for engineering problems, i.e. (Artificial) Neural Network, Gaussian Process, and Kriging. APPENDIX B also discusses other concepts such as statistical inferences, the problem of “Curse of Dimensionality”, the problem of regression, the regression related decision principles (i.e. the Empirical Risk Minimization principle, the principle of “Occam’s Razor”, and the Structural Risk Minimization principle).

2.3.1 Surrogate-Modeling Preliminaries

Surrogate-modeling methods are developed from statistical inference and regression estimation methods. In this section, the notion of surrogate model is discussed, and the relationship of statistical inference and regression estimation to surrogate-modeling methods is discussed.

2.3.1.1 The Notion of Surrogate Model

Kleijnen defines surrogate model (or metamodel) as a “model of a model” [32].

The two ‘models’ in Kleijnen’s definition have different meanings. The second ‘model’ means a physics-based mathematical model abstracting the mechanism of a physical phenomenon in a scientific and engineering domain. Usually a computer program or model can be established based on this mathematical model [33]. These physics-based computer models are accurate and comprehensive enough such that the process can be simulated for the corresponding physical phenomena and satisfactory

analysis results can be obtained with these computer programs. The significance of these achievements is obvious: both expensive experiments and time are greatly reduced if a similar phenomenon is to be studied. In this sense, the second “model” also means the corresponding computer analysis or simulation program.

In general, physics-based models work as follows: supplying a vector of design variables (inputs) X and computing a response (output) y . Thus, physics-based models can be represented as

$$y = f(X) \quad (2.12)$$

The first ‘model’, i.e. ‘surrogate model’, is an approximated model of the previous physics-based (computer) model and replaces the later one in the design process, especially in the conceptual design stage.

While time is a concern as discussed in Chapter 1, it is also very hard for the designer to get insight into the physics-based disciplinary models, because the designer often has an executable computer code instead of the source code. The situation is even worse in a multidisciplinary design. Thus, the designer may never uncover the functional relationship between design variables X and responses Y , and may never find the ‘best’ settings for design variables X [34].

Therefore, the multidisciplinary nature of the design of modern complex systems has posed challenges to the designers – how to decrease the time needed for a complete physics-based multidisciplinary analysis or simulation and get some insight into the functional relationship between design variables X and response y ?

A widely used strategy is to utilize approximation models, i.e. surrogate models, which are approximations of the complex physics-based models, but at a much lower cost

in terms of both time and computational load. This model is created by fitting a regression of some output values (i.e. the values of the responses) of the physics-based model, and these values are selected by some sampling techniques such as Design of Experiment (DoE). Because the surrogate model can not duplicate all the output values of the physics-based model, it is an approximation.

A surrogate model is often represented as

$$\hat{y} = g(X) \quad (2.13)$$

And so

$$y = \hat{y} + \varepsilon \quad (2.14)$$

where ε represents the error of approximation and/or (random) measurement errors, if any.

In order to substitute the original accurate and complex physics-based models, the surrogate models need to satisfy the following requirements:

- 1) Accurate enough, in order to obtain reliable prediction and subsequent design;
- 2) Much faster, justifying existence;
- 3) Easy to use, without complex setup work or many human interactions;

Besides, there are two additional requirements [34]:

- 4) Provide a better understanding of the functional relationship between design variables X and response y ;
- 5) Make easier integration of disciplinary models or surrogate models.

Now surrogate-modeling can be defined as the process of selection of an experimental design, a regression technique or surrogate model type, regression of the selected output values, and validation to assess the goodness of model fitting, to build a

“model of a model” as an approximate yet fast surrogate for a complex computer analysis or simulation program or code [35, 36].

However, a surrogate model is not limited to be a model of a physics-based model; it can be a model of a set of experimental data for which the physics-based model has not yet been established. In this case, obviously, it is nothing more than a regression model of the experimental data.

In general, the surrogate model is only meaningful in the predictive sense, while the physics-based model is both predictive and explanatory to the original physical phenomena. However, with certain regression techniques such as the Response Surface Methodology [33], the surrogate model is also meaningful in the explanatory sense if the contribution or importance of each factor or interaction in it is considered.

As an approximation, the goodness of model fitting and predicting accuracy of the surrogate models are important. However, in certain cases, part of the predicting capability of the surrogate model has to be sacrificed in order to obtain insights into the nature of the problem. One such example is the screening test. In a screening test, the main purpose is to identify the primary contributors to a response, and the goodness of model fitting and predicting accuracy are the second important concerns [37].

It should be pointed out that surrogate models are not only used to provide fast approximations for the original physics-based models, but also used to provide fast analyses for derivatives of the original physics-based models to reduce the computational cost for optimization [38, 39].

2.3.1.2 Surrogate-Modeling Based on Regression Estimation and Statistical Inference

Both surrogate-modeling and regression estimation try to obtain a mathematical relationship (function) between a response variable y and an input variable vector X . For this reason one may think surrogate-modeling is the same as regression estimation before looking in depth. Although surrogate-modeling is developed from regression estimation, these two methods are different in several aspects. First, the responses of sample data for surrogate-modeling have no random components for a given design variable vector X because those responses are generated by the computer program of the deterministic function between a response y and the design variable vector X , whereas those for regression estimation do because usually those responses are observed results of real life phenomena. Second, to generate the sample data, surrogate-modeling runs the computer program just once, whereas regression estimation needs to run the same DoE several times, or a distribution function for values of the response y has to be assumed if the DoE is run only one time because it is to obtain a mathematical relationship between the mean or expected value of a response variable y and a vector X of predictor variables (see Figure B-1 in APPENDIX B for example). Third, for surrogate-modeling it is better to use the modern space filling sampling techniques (modern DoE's). For regression estimation it is better to use the classical DoE's, as discussed previously. However, despite these differences, surrogate-modeling directly or indirectly uses the methods of regression estimation. Surrogate-modeling can directly use the methods developed for regression estimation to obtain the simpler approximated functional relationship because a deterministic response is a special case of the random ones described by distribution

functions. Except for the Gaussian Process, almost all surrogate-modeling methods make use of the concept of the empirical risk function or an equivalent concept such as mean square error. For these reasons, surrogate-modeling and regression estimation are usually thought of as the same. Surrogate-modeling also directly or indirectly uses the methods of statistical inference. For example, Kriging and Gaussian Process need to first directly infer the parameters of the assumed distribution function and then the surrogate model is established; other surrogate-modeling methods such as RSM need to check if the error distribution is close to the one implied or assumed by the regression method used and estimation of the error distribution uses the statistical inference methods.

Surrogate-modeling is related to regression estimation and statistical inference by **rephrasing** the Equation 2.14.

$$y = \hat{y} + \varepsilon \quad (2.14)$$

where error ε is **now** a systematic error related to the selection of the surrogate model \hat{y} .

The surrogate model \hat{y} is obtained by regression estimation methods; then the distribution of the error ε is analyzed by statistical inference methods; if the error distribution is close to the one implied or assumed by the regression method used, then one can be sure that a good surrogate model is obtained; otherwise, improvements to the surrogate model \hat{y} must be made until the requirements of goodness of fitting are met. One example is the RSM. A good surrogate model by RSM is obtained when the error distribution follows a normal distribution $N(0, \sigma^2)$ [33]; otherwise, means such as adding higher order terms (HOT) or transformation have to be taken to meet this requirement. Some surrogate-modeling methods need to first assume the properties or

distribution function of the error ε , and then this information is used to obtain the surrogate model. Such examples are Kriging and Gaussian Process.

The above cases can be called particular surrogate-modeling, because those methods make use of particular or parametric inference methods. Another class of surrogate-modeling methods is general surrogate-modeling, which does not need the information about the error distribution. Such examples are SVR and ANN.

Usually, if the functional relationship between the response and the design vector is known and simple to be described explicitly, particular surrogate-modeling should be used, such as the univariate linear regression method (see Figure B-1 in APPENDIX B for details). However, if many aspects of the physical phenomenon are unknown or hard to be described explicitly, such as the relationship embodied in a very complicated computer model, the general surrogate-modeling methods should be used as those methods are more versatile and powerful [40].

As a consequence of parametric inference, the particular surrogate-modeling methods suffer the problem of “curse of dimensionality” (see APPENDIX B for details), i.e. the sample size and computer resources have to be increased exponentially with the number of the design variables, or the model accuracy level increases slowly with the sample size. Besides, the accuracy of the results obtained by particular surrogate-modeling methods can be very bad if the assumed error distribution is far from the real one.

Because of independence on (error) distribution, the general surrogate-modeling methods do not have the problem of “curse of dimensionality”; and can obtain

satisfactory accuracy for many kinds of problems. Therefore, general surrogate-modeling methods have become popular in recent applications.

According to the form of the family of functions $g(X, \theta)$ (see APPENDIX B) during the regression process, the surrogate-modeling methods can be divided into another two classes: linear or nonlinear surrogate-modeling. For linear surrogate-modeling methods, such as RSM, the function family is the combinations of some definite functions (such as x , x^2 , $\sin x$, $\sin(2x)$, e^x , and e^{2x}) with θ being the coefficients of these definite functions, and the linear algebra method are often used to solve for all these coefficients. For nonlinear surrogate-modeling methods, such as ANN, SVR, the function family is a combination of some indefinite functions (i.e. function families, such as $\sin(ax+b)$, and e^{ax+b}) with θ being the coefficients of these indefinite functions and the unknown scalar(s) in these indefinite functions, and thus linear algebra can not be used to solve for θ . The indefinite functions are often called the “kernel functions” or simply “kernels”. Usually the parameter(s) in the kernel are pre-specified or determined before the coefficients of the kernels.

2.3.2 Two State-of-the-Art Surrogate-Modeling Methods

The Response Surface Methodology, one popular surrogate-modeling method for engineering problems, is reviewed, and the newly developing Support Vector Regression is introduced. APPENDIX B describes other three popular surrogate-modeling methods for engineering problems, i.e. (Artificial) Neural Network, Gaussian Process, and Kriging.

2.3.2.1 Response Surface Methodology

Response Surface Methodology is a well investigated and commonly applied surrogate-modeling method in engineering designs [33, 41]. In aerospace engineering,

RSM has been used for a variety of applications, particularly in multidisciplinary design and optimization [13, 42, 43]. RSM uses polynomials to approximate the true response behavior, and the polynomials are called the response surface equations (RSE). Usually a second order polynomial equation is used. The main reason for this is that considerable practical experience has shown that a second order model works well for many real problems. Higher order terms can be also added in if needed. The general second order RSE including linear, quadratic and interaction terms is as following:

$$\hat{y} = b_0 + \sum_{i=1}^n b_i x_i + \sum_{i=1}^n b_{ii} x_i^2 + \sum_{i=1}^{n-1} \sum_{j=i+1}^n b_{ij} x_i x_j \quad (2.15)$$

where \hat{y} is the predicted response, x_i are the design variables, b_0 is the intercept term, b_i , b_{ii} , and b_{ij} are related coefficients. b_0 , b_i , b_{ii} , and b_{ij} are the parameters to be estimated from the sample, and there are totally $(n+1)(n+2)/2$ parameters.

The maximum likelihood method is the general way to estimate the above parameters for any distribution the errors may have, but the least square approach is the easiest way if the errors are normally distributed [33]. Let

$$M_s = \begin{bmatrix} x_1 & \cdots & x_n & x_1^2 & \cdots & x_n^2 & x_1 x_2 & \cdots & x_{n-1} x_n \\ 1 & x_{11} & \cdots & x_{1n} & x_{11}^2 & \cdots & x_{1n}^2 & x_{11} x_{12} & \cdots & x_{1(n-1)} x_{1n} \\ 1 & x_{21} & \cdots & x_{2n} & x_{21}^2 & \cdots & x_{2n}^2 & x_{21} x_{22} & \cdots & x_{2(n-1)} x_{2n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{s1} & \cdots & x_{sn} & x_{s1}^2 & \cdots & x_{sn}^2 & x_{s1} x_{s2} & \cdots & x_{s(n-1)} x_{sn} \end{bmatrix},$$

$$Y_s = [y_1 \quad y_2 \quad \cdots \quad y_s]^T, \text{ and } B_s = [b_0 \quad b_1 \quad \cdots \quad b_n \quad b_{11} \quad \cdots \quad b_{nn} \quad b_{12} \quad \cdots \quad b_{(n-1)n}]^T,$$

then from the sample $S_s : \{(y_1, X_1), (y_2, X_2), \dots, (y_s, X_s)\}$ a least square estimation of the parameters is

$$B_s = (M_s^T M_s)^{-1} M_s^T Y_s \quad (2.16)$$

In fact, this polynomial approximation can be considered as a truncated Taylor series expansion around a point [41] with all higher order effects being negligible. Therefore, the resulting surrogate model may have poor approximation accuracy if there are many design variables and/or large ranges for the design variables, because the error term in the Taylor series expansion increases with the number of variables and the ranges of the variables [37]. If the behavior of the response is far away from a second order polynomial (or equivalently the error distribution is not normal), the accuracy will be poor, and higher order terms should be added in.

2.3.2.2 Support Vector Regression

In this section, the concept of Vapnik-Chervonenkis (VC) dimension and the structural risk function, which are the key parts of the theory of SVM, will be introduced, and then the theory of SVR is introduced.

Support Vector Regression (SVR) is a new surrogate-modeling method that originates from Support Vector Machine (SVM). SVM was developed for classification or pattern recognition problems starting in the late 1970s [44-46]. For classification or pattern recognition, SVM has been applied to many real world problems, such as isolated handwritten digit recognition, object recognition, speaker identification, face detection in images, and text categorization. SVM is also applied to several other areas, such as bio-informatics and artificial intelligence. Many other methods, such as ANN, have been used in these areas; however, what distinguishes SVM is its solid mathematical foundation: instead of adopting the empirical risk minimization (ERM) principle to minimize the empirical risk for a given sample, SVM adopts the SRM principle to

minimize the structural risk that is the upper bound of the (empirical) risk [44]. By minimizing this upper bound, SVM leads to lower model prediction errors for new or unseen data, i.e. those that are not in the sample, and thus has much better ability to generalize problems.

The main difference between a regression (or surrogate-modeling) problem and a classification problem is that the response of the regression problem is a continuous variable, whereas the response of the classification problem is a discrete variable with values such as -1 and 1. By replacing the loss function of the SVM method with a new one, the SVM method is modified to become the SVR method. Therefore, SVR inherits most of the advantages of SVM, such as the advantages of the SRM principle. It should be noted that this new risk function for SVR is not guaranteed to be the upper bound of the empirical risk because the response values of the sample may be far from the local extremes. This shows the importance of sampling methods that can be used to mitigate or even eliminate this problem. SVR also has other advantages. The optimization problem used to find the parameters of the surrogate model is a convex quadratic one, and thus there is only a global optimal solution and no other local minima. Because of this, SVR does not require a computation-intensive global optimizer and the resulting final model has high certainty, where certainty means that the final model should have similar performance if different algorithms or different search starting points are used to estimate the parameters in the surrogate model. Since there is only one (global) optimal solution for SVR, as long as the optimization method used by SVR can find this solution, the same final model will be obtained, and thus SVR has high certainty. The model fitting speed of SVR is relatively fast, usually taking less than one minute for a moderate sample

size. And it has the property of sparseness in sample point selection for the final surrogate model [47], where sparseness means that not all sample points are needed for the final surrogate model, although this can be known only after the model fitting process. The sample points that are used in the final surrogate model are called “support vectors”, and this is where SVM and SVR get the names and are distinguished from other methods.

Because of the above advantages, SVR has become the-state-of-art method for surrogate-modeling in recent years. SVR is not only applied to surrogate-modeling [35, 48, 49], but also many other areas such as time series prediction problems [50, 51], stock market prediction [52], and electricity load forecasting [53]. In these applications, SVR has shown promising empirical performance. Fan et al [49] is one of the first to apply SVR in the aerospace industry for regression fitting of aerodynamic data, in which SVR is concluded to have evidently better model predicting performance than ANN and another superiority over ANN: high certainty. In contrast, quite different final models may be obtained with different training algorithms for ANN, and thus ANN has lower certainty.

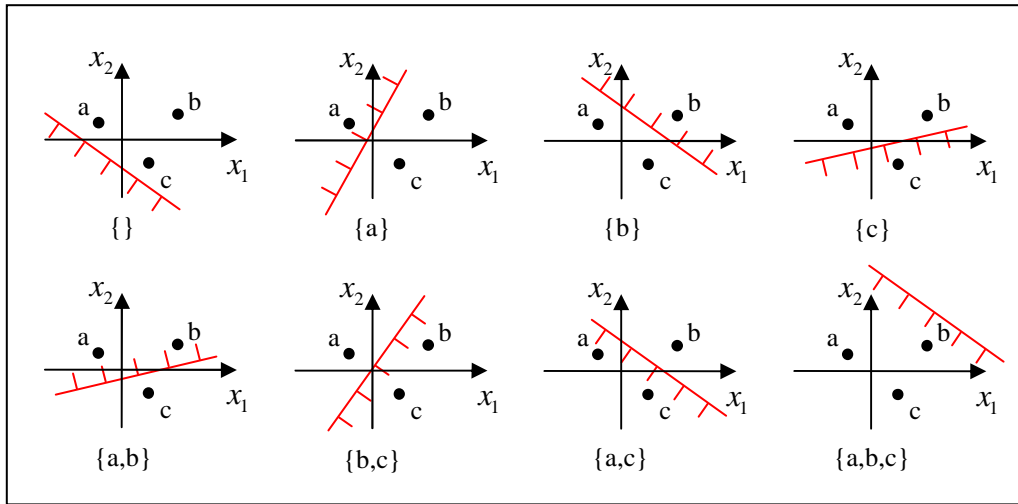
2.3.2.2.1 The VC Dimension and Structural Risk Function

SVM is formulated to solve a classification problem: given a (training) sample $S_s : \{(y_1, X_1), (y_2, X_2), \dots, (y_s, X_s)\}$, where $X = [x_1, x_2, \dots, x_n]^T$ is the vector of the (input) design variables, $y_i \in \{-1, 1\}$ (i.e. the value of y_i is either -1 or 1), find a hyperplane $g(X) = 0$ such that it separates the sample points X_i 's with $y_i = -1$ from those with $y_i = 1$ in the input space defined by x_1, x_2, \dots, x_n . In other words, find an approximation $\hat{y} = g(X)$, then the sign of \hat{y} can indicate which class this sample point X is in: the class of $y = -1$ or the class of $y = 1$.

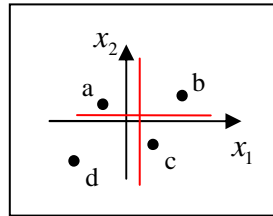
The separating hyperplane function family $g(X, \theta)$ has a Vapnik-Chervonenkis dimension. VC dimension is a scalar to measure the separating capacity of a separating/shattering hyperplane family. It is defined as follows [48, 44, 46]:

The VC dimension of a function family is h if and only if there exists a set of sample/training points $\{X_i\}_{i=1}^h$ such that these points can be shattered in all 2^h possible ways by this function family, and that no such set $\{X_i\}_{i=1}^q$ exists, where $q > h$, that also satisfies this property.

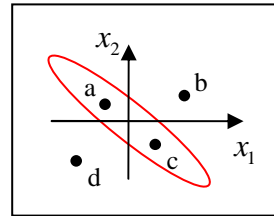
The process to determine the VC dimension of a linear function family $x_2 = ax_1 + b$ in the 2 dimensional input space is depicted in Figure 2-5. The 3 sample points a, b, and c can be separated by the linear functions (in red) in $2^3 = 8$ ways, Figure 2-5-A. However, the linear functions can not separate points a and c from points b and d if a fourth sample point d is added in, as depicted in Figure 2-5-B. Therefore, the VC dimension of this linear function family is 3 for these sample points. Figure 2-5-C shows a closed curve function can separate points a and c from points b and d.



A



B



C

Figure 2-5: Example to Show the VC Dimension of a Linear Function Family in the 2 dimensional Input Space [54]

Generally the VC dimension is not equal to the number of the free parameters in the function family. For example, the linear function family in the n dimensional input space has a maximum VC dimension of $(n+1)$, while the function family $x_2 = a \sin(bx_1) + c$ has a maximum VC dimension of infinite [55].

Use the concepts of risk function and empirical risk function, i.e. Equation B.5 and B.6 respectively (see APPENDIX B), the following bounds for the risk function holds with probability $1-\alpha$ [44],

$$R(g(X, \theta)) \leq R_{emp}(g(X, \theta)) + \sqrt{\frac{h \ln(2s/h + 1) - \ln(\alpha/4)}{s}} \quad (2.17)$$

where h is the VC dimension of the function family $g(X, \theta)$ for the give sample points, and s is the number of sample points.

Then the SVM model is constructed by solving the following problem [44]

$$\text{Minimize}_{\theta} : R_s = R_{emp}(g(X, \theta)) + \sqrt{\frac{h \ln(2s/h+1) - \ln(\alpha/4)}{s}} \quad (2.18)$$

where R_s is the structural risk function.

Therefore, SVM finds the optimal approximation $\hat{y} = g(X)$ by using the SRM principle to minimize the structural risk R_s that is the upper bound of the empirical risk. However, it is difficult to calculate the VC dimension h given specific sample points. The common practice is to find an upper bound on h and try to minimize this upper bound on h [46]. Therefore usually the practical form of the structural risk function R_s is different from the above Equation 2.18, as can be seen later.

2.3.2.2.2 The Theory of Support Vector Regression

The Support Vector Regression method is developed from the SVM method. At first, one may doubt how this can happen, since SVM is used to separate two groups of points while SVR is used to find an approximation function of the design variables to the response. Actually, SVM and SVR have one commonality: separating two groups of points, as shown as an example in Figure 2-6. In Figure 2-6-A the response values y_i of the red and green points are different, say -1 and 1, respectively, and the points are divided into two groups according to the response values. The goal of SVM is to find an optimal line to separate these two groups of points, and this line separates the two groups of points onto its two sides. In Figure 2-6-B, the points are generated from a linear relationship and an additive noise. The goal of SVR is to find an optimal line that

approximates the real linear relationship, and this line has also the effect of separating the sample points onto its two sides.

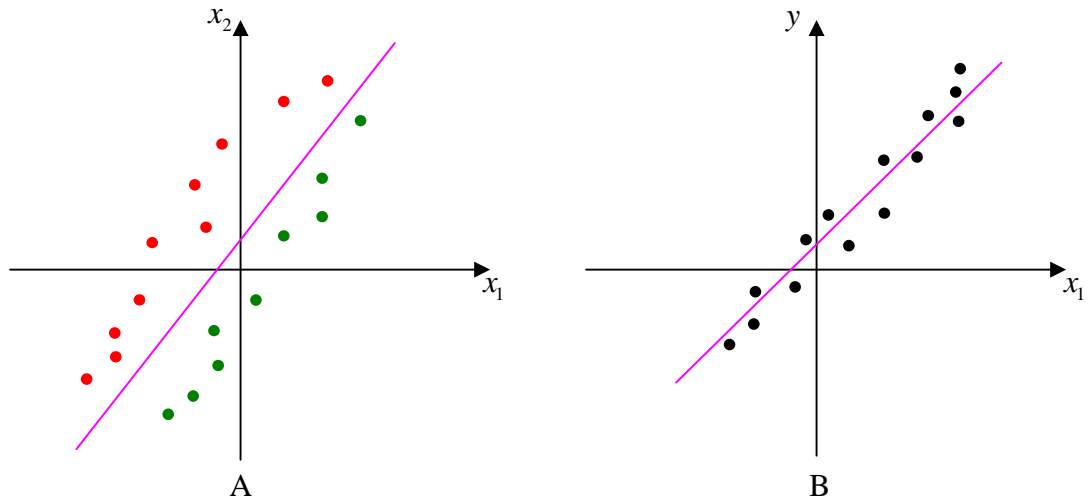


Figure 2-6: Example to Show the One Commonality between Classification and Regression

As mentioned previously, the SVM method can not be directly used for regression, its loss function should be replaced with a new one to consider the difference between classification and regression problems: the response values of classification are discrete, whereas those of the regression are continuous. In the rest of this section, popular loss functions of SVR are introduced, then the practical form of the structural risk function R_{ζ} is described, and finally the algorithm of SVR is provided.

There are four popular loss functions for SVR: quadratic, Laplace, Huber, and ε -insensitive loss functions. Other loss functions are also proposed but not popular, such as soft insensitive loss function [56], polynomial, piecewise polynomial [57], etc. Figure 2-7 illustrates the first four popular loss functions.

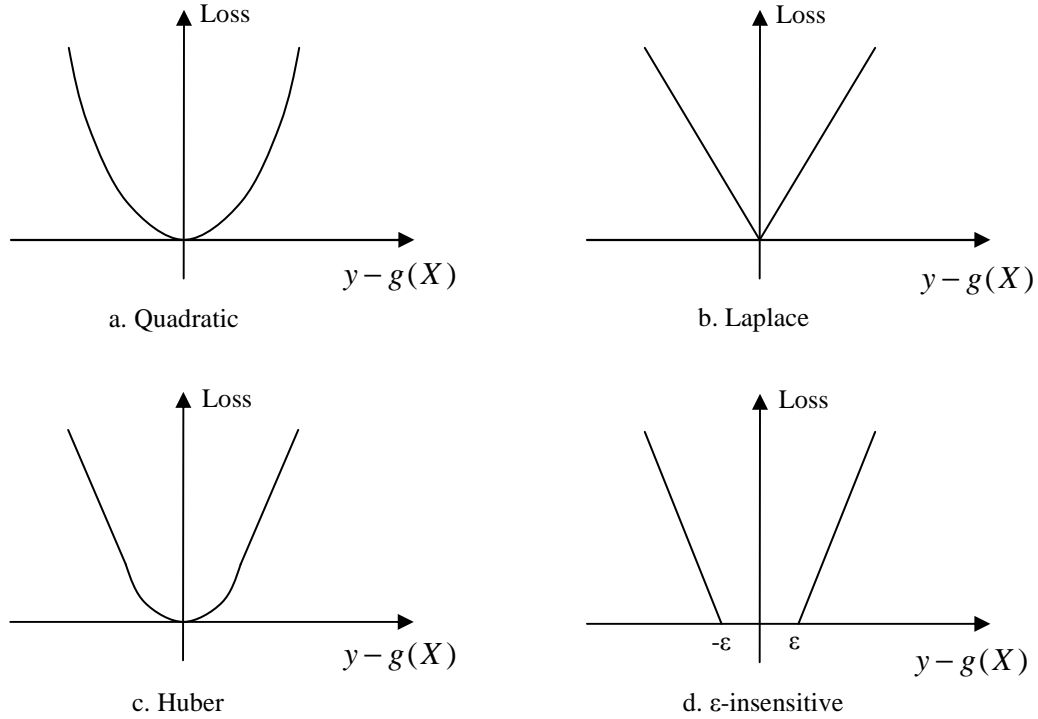


Figure 2-7: Illustrations of Four Loss Functions for SVR

The quadratic loss function is given as Equation 2.19, Laplace as Equation 2.20, Huber as Equation 2.21, and ε -insensitive as Equation 2.22.

$$L_{quad}(X, y, g(X)) = L_{quad}(y - g(X)) = (y - g(X))^2 \quad (2.19)$$

$$L_{Laplace}(X, y, g(X)) = L_{Laplace}(y - g(X)) = |y - g(X)| \quad (2.20)$$

$$L_{Huber}(X, y, g(X)) = L_{Huber}(y - g(X)) = \begin{cases} \frac{1}{2}(y - g(X))^2, & \text{if } |y - g(X)| < \mu \\ \mu|y - g(X)| - \frac{\mu^2}{2}, & \text{otherwise} \end{cases} \quad (2.21)$$

$$L_{\varepsilon}(X, y, g(X)) = L_{\varepsilon}(y - g(X)) = \begin{cases} 0, & \text{if } |y - g(X)| < \varepsilon \\ |y - g(X)| - \varepsilon, & \text{otherwise} \end{cases} \quad (2.22)$$

The performance of SVR depends on the loss function used [48, 57]. The quadratic loss function corresponds to the conventional least square error criterion and can be used if the errors are assumed to be normally distributed. The Laplace loss function is less

sensitive to outliers than the quadratic one. The Huber loss function is a robust loss function that has optimal properties when the error distribution is unknown. The ε -insensitive loss function is an approximation of Huber loss function and can also be used when the error distribution is unknown. The quadratic, Laplace and Huber loss functions will produce no sparseness in the sample points, i.e. all sample points will be used in the final regression model/surrogate model. The ε -insensitive loss function designed by Vapnik, however, can produce sparseness [55]. Therefore, due to this advantage and its calculation simplicity, the ε -insensitive loss function becomes the most frequently used loss function for SVR. However, one has to make a tradeoff between accuracy and sparseness if the ε -insensitive loss function is to be used [57] since less sparseness or more points used usually results in higher accuracy.

For SVR, the regression/surrogate model has the following form [58]:

$$\hat{y} = g(X) = \langle W, \Phi(X) \rangle + b \quad (2.23)$$

where $\langle \cdot, \cdot \rangle$ means dot product, W is a vector of scalars (weights) to be estimated, b is the bias or intercept to be estimated, and $\Phi(X)$ is a function vector. Therefore, the parameters to be estimated are W and b . The functions in $\Phi(X)$ can be linear or nonlinear, such as $\Phi(X) = (x_1, x_2)$ and $\Phi(X) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$, where $X = [x_1, x_2]^T$, and the explicit form of $\Phi(X)$ does not need to be known.

Based on the above denotations in Equation 2.23, the practical form of the structural risk function R_S to be minimized is given as

$$R_S = \frac{1}{s} \sum_{i=1}^s L(X_i, y_i, g(X_i)) + \frac{1}{2} \|W\|^2 \quad (2.24)$$

where $L()$ is the loss function, $\|W\|^2 = \langle W, W \rangle$. The first term on the right hand side is actually the empirical risk function.

The term $\frac{1}{2}\|W\|^2$ is related to the upper bound of the VC dimension of the functional family $g(X, \theta)$ as shown in the theory of SVM [48]. Because the VC dimension is a measure of the ‘‘capacity’’ of the functional family to approximate, and usually high capacity leads to overfitting [57], this term needs to be minimized. To minimize $\frac{1}{2}\|W\|^2$ is called to enforce ‘‘flatness’’ [58].

However, it is not easy to minimize the structural risk R_s given in Equation 2.24. It is found that the optimization problem of minimizing this structural risk can be converted to another convex minimization problem that is easier to solve and has only one global optimal solution. This alternative minimization problem is established as follows [57].

Let (residual) $\gamma = y - g(X)$, then the loss function can be written as $L(\gamma)$. Then the alternative convex quadratic minimization problem is given as

$$\begin{aligned} & \underset{W, b, \xi_i^+, \xi_i^-}{\text{Minimize}} && \frac{1}{2}\|W\|^2 + C \sum_{i=1}^s (L(\xi_i^- + \varepsilon) + L(\xi_i^+ + \varepsilon)) \\ & \text{subject to} && \begin{cases} y_i - \langle W, \Phi(X_i) \rangle - b \leq \varepsilon + \xi_i^+ \\ \langle W, \Phi(X_i) \rangle + b - y_i \leq \varepsilon + \xi_i^- \\ \xi_i^+, \xi_i^- \geq 0 \end{cases} \end{aligned} \quad (2.25)$$

where $C > 0$ is a pre-specified constant and sometimes called the regularization factor, ξ_i^+ and ξ_i^- are slack variables representing upper and lower bounds of the deviation $|y_i - g(X_i)|$, and $\varepsilon \geq 0$ is the pre-specified tolerable deviation.

The reason to introduce the slack variables ξ_i^+ and ξ_i^- is that the tolerable deviation ε can occasionally not be satisfied and thus some loss is incurred. ε has the same meaning as that in the ε -insensitive loss function, and for other loss functions $\varepsilon = 0$. Figure 2-8 depicts these variables when the ε -insensitive loss function is used for a linear regression problem.

The constant C determines the tradeoff between flatness of the final surrogate model $g(X)$ and the amount up to which deviations even greater than ε are accepted. If $C = \frac{1}{s}$, the alternative minimization problem, i.e. Equations 2.25, is exactly the same as the original problem of minimizing the structural risk given in Equation 2.24. With an increase of C more emphasis is put on the loss function and the structural risk function Equation 2.24 is more like the conventional empirical risk function, with less regard to capacity of the functional family. However, the value of C can be optimized, and thus the alternative minimization problem is more flexible than the original problem.

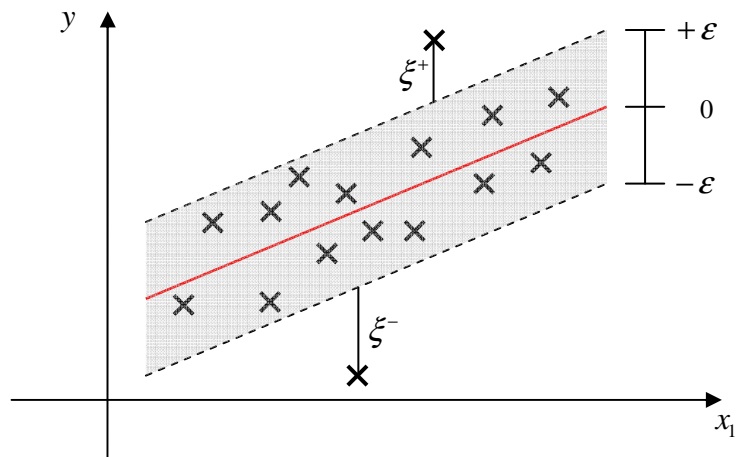


Figure 2-8: Example to Account for Slack Variables for Linear Regression with the ε -Insensitive Loss Function

The Lagrangian function of Equations 2.25 is constructed as [59]

$$\begin{aligned}
F_{Lag} = & \frac{1}{2} \|W\|^2 + C \sum_{i=1}^s (L(\xi_i^- + \varepsilon) + L(\xi_i^+ + \varepsilon)) \\
& - \sum_{i=1}^s \alpha_i^+ (\varepsilon + \xi_i^+ - y_i + \langle W, \Phi(X_i) \rangle + b) \\
& - \sum_{i=1}^s \alpha_i^- (\varepsilon + \xi_i^- + y_i - \langle W, \Phi(X_i) \rangle - b) \\
& - \sum_{i=1}^s \eta_i^+ \xi_i^+ - \sum_{i=1}^s \eta_i^- \xi_i^-
\end{aligned} \tag{2.26}$$

where F_{Lag} is the Lagrangian function, α_i^+ , α_i^- , η_i^+ and η_i^- are Lagrangian multipliers, and W , b , ξ_i^+ and ξ_i^- are called the primal variables.

The multipliers must be non-negative, i.e.

$$\alpha_i^+, \alpha_i^-, \eta_i^+, \eta_i^- \geq 0 \tag{2.27}$$

According to the necessary conditions for the saddle point of the Lagrangian function to be an optimal solution of its original problem Equation 2.25, the partial derivatives of F_{Lag} with respect to the primal variables have to vanish, i.e.

$$\partial_W F_{Lag} = W - \sum_{i=1}^s (\alpha_i^+ - \alpha_i^-) \Phi(X_i) = 0 \tag{2.28}$$

$$\partial_b F_{Lag} = \sum_{i=1}^s (\alpha_i^- - \alpha_i^+) = 0 \tag{2.29}$$

$$\partial_{\xi_i^+} F_{Lag} = C \partial_{\xi_i^+} L(\xi_i^+ + \varepsilon) - \alpha_i^+ - \eta_i^+ = 0 \tag{2.30}$$

$$\partial_{\xi_i^-} F_{Lag} = C \partial_{\xi_i^-} L(\xi_i^- + \varepsilon) - \alpha_i^- - \eta_i^- = 0 \tag{2.31}$$

Substituting Equations 2.28, 2.29, 2.30, and 2.31 into 2.26 yields the convex quadratic dual optimization problem [59], omitting some superscripts + and -, and some subscript i of α and ξ , as follows:

$$\begin{aligned}
& \text{Maximize}_{\alpha_i^+, \alpha_i^-} \left\{ \begin{aligned} & -\frac{1}{2} \sum_{i,j=1}^s (\alpha_i^+ - \alpha_i^-)(\alpha_j^+ - \alpha_j^-) \langle \Phi(X_i), \Phi(X_j) \rangle \\ & + \sum_{i=1}^s y_i (\alpha_i^+ - \alpha_i^-) - \sum_{i=1}^s \varepsilon (\alpha_i^+ + \alpha_i^-) \\ & + C \sum_{i=1}^s (T(\xi_i^+ + \varepsilon) + T(\xi_i^- + \varepsilon)) \end{aligned} \right. \quad (2.32) \\
& \text{subject to} \left\{ \begin{aligned} & \sum_{i=1}^s (\alpha_i^+ - \alpha_i^-) = 0 \\ & \alpha \leq C \partial_{\xi} L(\xi + \varepsilon) \\ & \xi = \inf \{ \xi \mid C \partial_{\xi} L(\xi + \varepsilon) \geq \alpha \} \\ & \alpha, \xi \geq 0 \end{aligned} \right.
\end{aligned}$$

where

$$\begin{cases} \bar{W} = \sum_{i=1}^s (\alpha_i^+ - \alpha_i^-) \Phi(X_i) \\ T(\xi + \varepsilon) = L(\xi + \varepsilon) - \xi \bar{\partial}_{\xi} L(\xi + \varepsilon) \end{cases} \quad (2.33)$$

Denoting $k(X_i, X_j) = \langle \Phi(X_i), \Phi(X_j) \rangle$, which is called the kernel function, the optimization problem Equations 2.32 is equivalent to the following standard optimization problem:

$$\begin{aligned}
& \text{Minimize}_{\alpha_i^+, \alpha_i^-} \left\{ \begin{aligned} & \frac{1}{2} \sum_{i,j=1}^s (\alpha_i^+ - \alpha_i^-)(\alpha_j^+ - \alpha_j^-) k(X_i, X_j) \\ & - \sum_{i=1}^s y_i (\alpha_i^+ - \alpha_i^-) + \sum_{i=1}^s \varepsilon (\alpha_i^+ + \alpha_i^-) \\ & - C \sum_{i=1}^s (T(\xi_i^+ + \varepsilon) + T(\xi_i^- + \varepsilon)) \end{aligned} \right. \quad (2.34) \\
& \text{subject to} \left\{ \begin{aligned} & \sum_{i=1}^s (\alpha_i^+ - \alpha_i^-) = 0 \\ & \alpha \leq C \partial_{\xi} L(\xi + \varepsilon) \\ & \xi = \inf \{ \xi \mid C \partial_{\xi} L(\xi + \varepsilon) \geq \alpha \} \\ & \alpha, \xi \geq 0 \end{aligned} \right.
\end{aligned}$$

By solving this optimization problem, Equations 2.34, the values of α and ξ are determined. Because this optimization problem is quadratic, it is convex and thus has only one global optimal solution.

The ε -insensitive loss function is used to show how to further simplify Equations 2.34 to make it practically useful. In this case, $L(\xi + \varepsilon) = |\xi| = \xi$. Then we get

$$T(\xi + \varepsilon) = \xi - \xi \cdot 1 = 0$$

Moreover, one can conclude from $\partial_{\xi} L(\xi + \varepsilon) = 1$, $\alpha \leq C \partial_{\xi} L(\xi + \varepsilon)$, and $\xi, \alpha \geq 0$ that $\alpha \in [0, C]$ and $\xi = \inf\{\xi \mid C \geq \alpha\} = 0$. Then the Equations 2.34 are simplified as

$$\begin{aligned} \text{Minimize}_{\alpha_i^+, \alpha_i^-} & \begin{cases} \frac{1}{2} \sum_{i,j=1}^s (\alpha_i^+ - \alpha_i^-)(\alpha_j^+ - \alpha_j^-) k(X_i, X_j) \\ - \sum_{i=1}^s y_i (\alpha_i^+ - \alpha_i^-) + \sum_{i=1}^s \varepsilon (\alpha_i^+ + \alpha_i^-) \end{cases} \\ \text{subject to} & \begin{cases} \sum_{i=1}^s (\alpha_i^+ - \alpha_i^-) = 0 \\ 0 \leq \alpha \leq C \end{cases} \end{aligned}$$

Table 1 summarizes the conditions on α and formulas of $CT(\xi + \varepsilon)$ for the four popular loss functions.

Table 1: Some Terms of the Dual Optimization Problem for Different Loss Functions

Loss function	ε	α	ξ	$\partial_{\xi}L(\xi + \varepsilon)$	$CT(\xi + \varepsilon)$
Quadratic	$\varepsilon = 0$	$\alpha \in [0, \infty]$	$\frac{\alpha}{2C}$	$\frac{\alpha}{C}$	$-\frac{\alpha^2}{4C}$
Laplace	$\varepsilon = 0$	$\alpha \in [0, C]$	0	1	0
Huber	$\varepsilon = 0$	$\alpha \in [0, C\mu]$	$\begin{cases} \frac{\alpha}{C}, & \text{if } \alpha < \mu C \\ \mu, & \text{otherwise} \end{cases}$	$\begin{cases} \frac{\alpha}{C}, & \text{if } \alpha < \mu C \\ \mu, & \text{otherwise} \end{cases}$	$\begin{cases} -\frac{\alpha^2}{2C}, & \text{if } \alpha < \mu C \\ -\frac{C\mu^2}{2}, & \text{otherwise} \end{cases}$
ε -insensitive	$\varepsilon \neq 0$	$\alpha \in [0, C]$	0	1	0

If the kernel function has a bias term, such as the inhomogeneous polynomial kernel shown later, then \bar{b} is accommodated within the kernel function as a result of the optimization process [48]. In this case, the term should be dropped and the surrogate model is given by

$$\hat{y} = g(X) = \sum_{i=1}^s (\alpha_i^+ - \alpha_i^-) k(X_i, X) \quad (2.35)$$

Observing Equation 2.35 it can be found that the input vectors of the design variables, i.e. the sample points X_i or the new point X , only appear inside the dot product of the kernel function $k(X_i, X_j) = \langle \Phi(X_i), \Phi(X_j) \rangle$. Because this dot product for any one pair of input vectors is a scalar, the dimensionality of the input space is hidden from the remaining optimization process. This provides a way of addressing the ‘‘curse of dimensionality’’ [48].

According to the KKT conditions, for the saddle point of the Lagrangian function to be an optimal solution of its original problem Equation 2.25, 3 sets of necessary conditions must be satisfied. The following are some of these conditions:

$$\begin{aligned}\alpha_i^+(\varepsilon + \xi_i^+ - y_i + \langle \bar{W}, \Phi(X_i) \rangle + b) &= 0 \\ \alpha_i^-(\varepsilon + \xi_i^- + y_i - \langle \bar{W}, \Phi(X_i) \rangle - b) &= 0\end{aligned}$$

If the ε -insensitive loss function is used, several useful conclusions can be drawn from the above equations. First, for the samples outside the ε -tube (see the shaded region in Figure 2-8 as an example), i.e. samples with $|y_i - g(X_i)| \geq \varepsilon$, it can be shown that $(\varepsilon + \xi_i^+ - y_i + \langle W, \Phi(X_i) \rangle + b) = 0$ or $(\varepsilon + \xi_i^- + y_i - \langle W, \Phi(X_i) \rangle - b) = 0$, but not both. For all the samples inside the ε -tube, i.e. $|y_i - g(X_i)| < \varepsilon$, it can be shown that $(\varepsilon + \xi_i^+ - y_i + \langle W, \Phi(X_i) \rangle + b) > 0$ and $(\varepsilon + \xi_i^- + y_i - \langle \bar{W}, \Phi(X_i) \rangle - b) > 0$. These conclusions mean that for the samples outside the ε -tube only one of the Lagrangian multipliers α_i^+ and α_i^- is nonzero and the other one is zero, and for all the samples inside the ε -tube the Lagrangian multipliers α_i^+ and α_i^- are zero. Second, from Equation 2.35 and the first conclusion it is known that not all samples contribute to the estimation of \bar{W} and the consequent surrogate model $g(X)$, but only the ones outside of the ε -tube do. This is the sparseness in sample selection, and the sample points with nonzero Lagrangian multipliers are called the “support vectors”.

The sparseness feature is very important when the sample is large because it reduces the number of terms in the surrogate model resulting in some loss of accuracy but improves the calculation speed of the surrogate model. Otherwise, the calculation can be

quite slow when the sample size is very large. For this unique advantage, the ϵ -insensitive loss function is selected as the only loss function for SVR in this research.

Not all functions can be selected as the kernel function $k(X, X')$, where X and X' are input vectors. The kernel function $k(X, X')$ has to satisfy Mercer's condition [57], such that the kernel matrix $K_{ij} = k(X_i, X_j)$ is positive definite in order that a unique optimal solution is guaranteed to the quadratic optimization problem Equation 2.34 [35]. Table 2 lists the common kernel functions.

Table 2: Common Kernel Functions of SVR

Linear	$k(X, X') = \langle X, X' \rangle$
Polynomial	$k(X, X') = \langle X, X' \rangle^d$
Inhomogeneous polynomial	$k(v) = (\langle X, X' \rangle + c)^d$, c is a constant (bias)
Gaussian radial basis function (GRBF)	$k(X, X') = \exp\left(-\frac{\ X, X'\ ^2}{2\sigma^2}\right)$
Exponential radial basis function (ERBF)	$k(X, X') = \exp\left(-\frac{\ X, X'\ }{2\sigma^2}\right)$
Sigmoid (multi-layer perceptron)	$k(X, X') = \tanh(\rho\langle X, X' \rangle + \gamma)$

A new kernel can be generated by positive linear combination of kernels, or from the product of kernels [57].

$$\text{Combination: } k(X, X') = \sum_i c_i k_i(X, X'), \quad c_i > 0$$

$$\text{Product: } k(X, X') = \prod_i k_i(X, X')$$

The Matlab[®] codes of SVR used in this research are developed based on the codes in Ref. [48].

2.3.3 Comparisons of Surrogate-Modeling Methods

Although many surrogate-modeling methods have been developed, only a small number of those methods have been successfully applied to various engineering design processes from different fields, such as Response Surface Methodology (RSM), Kriging (KG), Gaussian Process (GP), (artificial) neural network (ANN), radial basis functions (RBF), and multivariate adaptive regression splines (MARS).

However, even those successful surrogate-modeling methods have advantages and disadvantages, and there is no single method that is superior to the others in all circumstances. Some surrogate-modeling methods are very good at some particular types or domains of engineering problems, but those methods fail to achieve adequate performance for other types or domains of problems. Examples can be seen in the surrogate-modeling comparison literatures such as Ref. [35] and [60]. The reasons causing this phenomenon are the complex nature of engineering physics-based models and the performance of the SM's. For example, some SM's are good at low order nonlinear relationships but not good at high order ones, such as the second order RSM; on the other hand, some SMs are good at high order nonlinear relationships but not good at low order ones, such as MARS [60].

Different SM's can be compared qualitatively in a more theoretical way than observing SM's performance of handling engineering problems. Generally, the comparison criteria for different SM include the following [60]:

1. Accuracy for different complexity (order of nonlinearity) of test problems, under different sample sizes (scale of the sample data), and with noise;

2. Robustness in terms of variance of error values for different samples generated by different sampling methods;

3. Efficiency in terms of time used for surrogate model construction and new predictions;

4. Transparency in terms of the capability of providing information for contributions of factors and interactions among those factors;

5. Simplicity in terms of the number of parameters needed to be specified by a user.

In Ref. [60], four well known methods including RSM, MARS, RBF and KG are systematically compared with the above five criteria. There are functions of 14 test problems of different complexity in this comparison. The comparison results show the following:

1. In terms of accuracy and robustness, MARS, RBF and KG perform equally well under large sample sizes; RBF is the best under small and scarce sample size; and RSM is the best with noise;

2. In terms of efficiency for surrogate model construction, KG is the most time-consuming; and RSM needs the least time;

3. In terms of efficiency for new predictions, all methods need trivial time and work equally well;

4. In terms of transparency, RSM is the best in that a simple polynomial function is obtained and the contributions of each design variable and the interaction among those variables can be easily assessed;

























































5. In terms of simplicity, both RSM and RBF are the best in that the user does not need to specify any parameters to obtain the best accuracy, whereas MARS and KG need the involvement of the user to do so.






In Ref. [35], SVR is systematically and quantitatively compared with four well known methods including RSM, MARS, RBF, and KG with the previous four criteria for 26 engineering test problems of different complexity. The comparison results show SVR has the best overall performance, i.e. in terms of accuracy and robustness, SVR outperforms almost all the other four methods except for KG has smaller maximum error; in terms of efficiency, SVR takes similar time as MARS and much less time than KG, whereas RSM and RBF are much faster; in terms of transparency, SVR has explicit functions as RSM and RBF, but can not tell the contributions of each design variable and the interaction among those variables. However, in terms of simplicity, SVR needs the user to specify several parameters depending on the kernel used, and thus is not easy to use.

Although ANN is also a well known method and can approximate complex models very well, it is not included in the comparisons in Ref. [35] and [60] mainly due to three reasons [61]: first, it has no transparency in that it is hard to output and understand the functions that construct the surrogate model; second, how to obtain the best fitted model for a given training sample is still an art since there are many factors to be pre-selected, such as the number of layers and optimization algorithm for training; and third, it takes a long training time and expensive computational cost. On the other hand, it has been shown in many engineering applications that SVR produces equally accurate, if not better, results than ANN [35]. Moreover, it has been shown that ANN tends to overfit the

sample data and results in a model that is accurate with the sample data but has large errors with new predictions [48], whereas SVR has no such problem because of its mathematical foundation. Table 3 shows **qualitative** comparison of different surrogate-modeling methods.

Table 3: Qualitative Comparison of Surrogate-Modeling Methods

	Accuracy (RMSE)	Accuracy (MAE)	Robustness	Computing efficiency	Transparency	Simplicity	No Over-fitting	No Curse of dimensionality
RSM								
MARS								
RBF								
Kriging								
GP								
ANN								
SVR								

Excellent:  Good:  Fair:  Yes:  No: 

Note: RMSE means root mean square error; MAE means maximum absolute error;

Over-fitting and Curse of dimensionality are two disadvantages to be avoided;

RMSE and MAE are defined in APPENDIX D, and curse of dimensionality in APPENDIX B.

2.4 Model Assessment and Model Selection Methods

In this sub-section, the concepts of model assessment and model selection methods are described, and the information criteria are summarized. Then practical methods are introduced to select three parameters of SVR, i.e. the regularization factor C , ε if the ε -insensitive loss function is used, and the kernel parameter σ if GRBF or ERBF kernel function is used.

APPENDIX D discusses two types of model errors, i.e. model fitting error and model predicting error, and two popular measures of model errors, i.e. Root Mean Square Error (RMSE) and Maximum Absolute Error (MAE). APPENDIX D also summarizes other two popular model assessment and model selection methods, i.e. cross validation and bootstrap based on model predicting error, which are classified as re-sampling methods.

2.4.1 Concepts of Model Assessment and Model Selection

After construction of surrogate models, the quality of the resulted surrogate models should be assessed based on some criteria. In Ref. [60] multiple assessment criteria are advocated for assessment and comparison of surrogate-modeling methods, including accuracy, efficiency, robustness, model transparency, and simplicity. However, robustness and transparency are difficult to quantify; the computing efficiency is not a big concern for surrogate-modeling since all resulted surrogate models are fast for new predictions, although the construction efficiency is a concern before model selection; and simplicity can be addressed by the model fitting process (note that simplicity is concerned with the number of parameters that need to be specified by the user). Therefore, robustness, transparency, efficiency, and simplicity will not be used as criteria

for **quantitative** model selection. Accuracy seems now to be the only criterion for model selection. However, complexity should be another criterion, as implied by the Occam's razor principle (see APPENDIX B for details), that a simpler surrogate model is preferred. Low complexity is considered to generally imply low model predicting error because a simpler surrogate model is less likely to overfit a sample set. Overfitting is considered as the main cause of high model predicting error when the model fitting error is low. However, if a surrogate model is too simple for a problem, for example, a straight line for a circle, this simple model will also have high predicting error because of underfitting. Therefore, the criteria to assess surrogate models should include both accuracy and complexity. Accuracy can be measured by the model fitting error and/or the model predicting error. The complexity can be measured by the number of parameters to be estimated. Unfortunately, all existing model assessment methods do not use all of model fitting error, model predicting error, and model complexity.

Simply speaking, the problem of model selection is to select a surrogate model that best satisfies the given criterion from a set of surrogate models. It includes 2 or 3 folds: selection of surrogate model structures or surrogate-modeling methods; selection of parameters of the surrogate model; and if a kernel surrogate-modeling method is used, such as SVR, the selection of kernel functions. The surrogate model structure implies the specific form of the surrogate model assumed by a surrogate-modeling method, for example, the RSM assumes the polynomial functions as the surrogate model structure. Since different surrogate-modeling methods have different degrees of complexity, there is a need to select a simpler surrogate model structure on top of adequate accuracy for a specific problem. Although the parameters of a surrogate model are estimated based on

the given sample set, the estimation results of the parameters can be different if different estimation algorithms are used, such as the different training algorithms in the ANN surrogate model. Therefore, there is a need to select the best set of the estimation results for the parameters. For the kernel function based surrogate-modeling methods, different kernel functions result in different final surrogate models, and thus there is a need to select the best kernel function.

2.4.2 Information Criteria

Information criteria (IC) are one class of methods that are devised specifically for the purpose of model selection. Here, “devise” implies that the practical forms of information criteria are not derived mathematically (i.e. no proof), but given. However, those methods do have some theoretical foundations, such as the maximum likelihood principle. Additionally, IC methods have shown great success in model selection for a wide array of problems. For these reasons, the practical forms of the information criteria will be provided directly without detailed review of the theoretical foundations.

There are three main approaches of the information criterion methods: Akaike information criterion (AIC), Bayesian information criterion (BIC), and minimum description length (MDL). The best model is the one with the minimum value of the information criterion. The MDL [62] is not as popular as AIC and BIC because of three main reasons. First, the MDL uses description length as the information criterion, which is based on the coding theory of the information theory¹, but the extension of description

¹ Information theory is the mathematical theory of data communication and storage to tackle the engineering problem of the reliable transmission of information over a noisy channel. Its main result is that by appropriate encoding and decoding of the information, the information can be communicated over a noise channel with an arbitrarily small probability of error (http://en.wikipedia.org/wiki/information_theory)

length as a measure of the goodness of a model is not intuitive to many engineers. Second, the MDL is complicated for application because it does not have a fixed form for different model selection problems and thus the appropriate form for a specific family of models has to be derived accordingly. Third, its appropriate form for a model selection problem is often found to be almost the same as BIC. For these reasons, the MDL is not described in this research.

The AIC is the first IC method devised for general model selection problems [63]. It ingeniously incorporates two information sources: the goodness of model fitting and the complexity of a model, and achieves a balance between these two. The goodness of model fitting is measured by the log-likelihood function based on the maximum likelihood principle, or exactly the Kullback-Leibler information criterion [64]; and the complexity of a model is measured by the number of parameters of the model to be estimated. An equivalent but computationally convenient expression, i.e. the practical form, is given as [65, 66]

$$\text{AIC} = \ln(\hat{\sigma}_{\text{MLE}}^2) + \frac{2m}{s} \quad (2.36)$$

where m is the number of parameters in the model, s is the sample size, and $\hat{\sigma}_{\text{MLE}}^2$ denotes the maximum likelihood estimation of the variance of the residual term,

$$\hat{\sigma}_{\text{MLE}}^2 = \frac{\sum_{i=1}^s (y_i - \hat{y}_i)^2}{s} \quad (2.37)$$

The BIC is another popular information criterion [67]. It takes a Bayesian approach for model selection, deriving an approximation to a Bayesian posterior estimation of the parameters of a model from the given sample. A practical form of BIC is given as [65, 66]

$$\text{BIC} = \ln(\hat{\sigma}_{\text{MLE}}^2) + \frac{m \ln(s)}{s} \quad (2.38)$$

Comparing Equations 2.36 and 2.38 one can infer that the BIC imposes more penalty on model complexity than AIC if $s \geq 8$. Therefore, BIC will select a model with the number of parameters no greater than that selected by AIC. In addition, BIC is shown to select the correct model asymptotically with probability one² if the correct model is one of the candidate models and the sample size $s \rightarrow \infty$ [65]. For these reasons, BIC is often preferred to AIC for engineering applications that can only afford a small sample size [68].

Comparing Equations 2.36 and 2.38 one can also find that AIC and BIC can be generalized as the following form [66]

$$\text{IC} = \ln(\hat{\sigma}_{\text{MLE}}^2) + m\varphi(s) \quad (2.39)$$

where $\varphi(s)$ is a positive function of the sample size and satisfies the condition $\lim_{s \rightarrow \infty} \varphi(s) = 0$.

Based on this generalization, many derivatives of AIC and BIC are devised by modifying one or two terms in the generalized Equation 2.39 for better performance with respect to specific surrogate-modeling methods, such as the modified AIC and BIC for neural network in Ref. [68], the ones in Ref. [69], and the one in Ref. [70]. In this research, new modified AIC and BIC are devised to include three kinds of information: model fitting error, model complexity, and model predicting error.

² A model selection criterion that select the correct model asymptotically with probability one if the sample size approaches infinite is said to be consistent.

2.4.3 Practical Selection of Three SVR Parameters

There are many parameters to be selected or estimated in SVR. Those parameters are the weight vector W , the intercept b , the regularization factor C , the kernel parameter(s), and the tolerable deviation ε if the ε -insensitive loss function is used (for other loss functions $\varepsilon = 0$). Since the quality of the final surrogate model depends on all the parameters, now the question is: how can all these parameters be selected or determined? By solving the alternative convex quadratic minimization problem Equation 2.25, W and b can be determined, given pre-specified C , ε , and kernel parameter(s) that are usually given such as by experts of SVR. Now the question is reduced to: how can a non-expert user pre-specify or select C , ε , and kernel parameter(s)?

One way to pre-specify these three sorts of parameters is to use an optimizer, such as the Genetic Algorithms or Simplex Optimization as described in Ref. [71]; another way is to construct a new alternative minimization problem, such as described in Ref. [72] to automatically select ε given pre-specified C and kernel parameter(s). However, the two approaches are computationally expensive, and do not make use of the information contained in the sample, i.e. a priori knowledge, to select these parameters, as the SVR experts do.

The parameters C and ε can be selected based on the information contained in the sample no matter what the kernel function and kernel parameter(s) are. The kernel parameter(s) can be selected later using model selection methods discussed previously. Therefore here the focus is on the selection of the parameters C and ε , using the practical methods in Ref. [73].

When the ε -insensitive loss function is used, as mentioned previously, the regularization factor C determines the tradeoff between flatness of the final surrogate model $g(X)$ and the amount up to which deviations greater than ε are accepted; and with increase of C more emphasis is put on the loss function and the structural risk function in Equation 2.24 is more like the conventional empirical risk function. As described previously (see Table 1), in this case the dual variables $0 \leq \alpha \leq C$. Further, referring to Equation 2.35, the dual variables are used as linear coefficients in the final surrogate model. Therefore, a “good” value for C could be chosen to be equal to the range of the response values of a sample. However, this selection of C is sensitive to possible outliers in the sample, thus the practical selection of C is given as

$$C = \max\left(|\bar{y} + 3\sigma_y|, |\bar{y} - 3\sigma_y|\right) \quad (2.40)$$

where \bar{y} and σ_y are the mean and the standard deviation of the response values of the sample, respectively.

As shown previously (see Figure 2-8), the tolerable deviation ε controls the width of the ε -insensitive zone. In addition, according to the theory provided before the value of ε affects the number of support vectors to construct the final surrogate model. Therefore, the value of ε should be proportional to the model fitting error level. On the other hand, the selection of ε should depend on the sample size: intuitively, larger sample sizes should require smaller value of ε such that more support vectors can be selected to improve accuracy. Two practical selections of ε are given as

$$\varepsilon = 3\sigma_E \sqrt{\frac{\ln s}{s}}, \text{ or } \varepsilon = \frac{\sigma_E}{s}$$

where σ_E is the standard deviation of the residuals or model fitting error.

In this research it is found that the second practical selection can provide better results in most cases, and thus is used for selection of ε .

$$\varepsilon = \frac{\sigma_E}{\sqrt{s}} \quad (2.41)$$

Now the problem is that the residuals are not known a priori, and need to be estimated from the sample. The k -nearest-neighbor method can be used to estimate σ_E . In the k -nearest-neighbor method, the (pseudo) predicted response value of each sample point \tilde{y}_i is estimated as the average of the responses values of the k nearest sample points, where the distance between two sample points is measured by the Euclidian distance. Typically, the value of k is in the 2 – 6 range, and a value of 3 is recommended and used in this research. Then the estimation of σ_E is given as

$$\hat{\sigma}_E^2 = \frac{s^{\frac{1}{5}}k}{s^{\frac{1}{5}}k - 1} \frac{1}{s} \sum_{i=1}^s (y_i - \tilde{y}_i)^2 \quad (2.42)$$

Then the estimation $\hat{\sigma}_E$ of σ_E is substituted into Equation 2.41 to select the practical value of ε .

If the Gaussian radial basis function is selected as the kernel function, the kernel parameter σ , which is called the width of the radial basis function, is also suggested in Ref. [73] to be set to

$$\sigma^n \sim (0.1, 0.5)$$

where all the n design variables are pre-scaled to [0,1] range.

In this research it is found the interval can be extended to a wider one and sometimes provide better results.

$$\sigma^n \sim (0.01, 0.5) \quad (2.43)$$

2.5 Preliminaries of Probabilistic Design Methods

In the section, the basics of probabilistic design and the methods to incorporate the effects of uncertainties in design are summarized.

2.5.1 The Basics of Probabilistic Design

In early design stages there is a high degree of uncertainty. As mentioned in Chapter 1, uncertainty is the incompleteness of design knowledge, or a difference between reality and what is expected. There are many specific sources of uncertainties, such as ambiguity of the requirements, analysis or simulation tool fidelity, incomplete knowledge of the manufacture process and operational environment including human interactions, immaturity of the new technologies, and approximation errors introduced by the surrogate models for the physics-based analysis and simulation tools. Figure 2-9 provides a comprehensive summary of the sources of uncertainty and error in computational simulation in early design stages.

Although surrogate models are used to increase knowledge in the early design stages such that educated (with more information) decisions can be made and avoid locking in the final life cycle cost (LCC) and performance, it is still possible that bad decisions can be made because of the uncertainty existing in these stages. On the other hand, as mentioned in Chapter 1, the design solutions of the traditional deterministic multidisciplinary design optimization may be highly sensitive to the variation of the uncertain factors, leading to performance loss, or suffer from high likelihood of undesired events, or being conservative and consequently uneconomic. Therefore, advanced design techniques or methods have been developed to handle uncertainties.

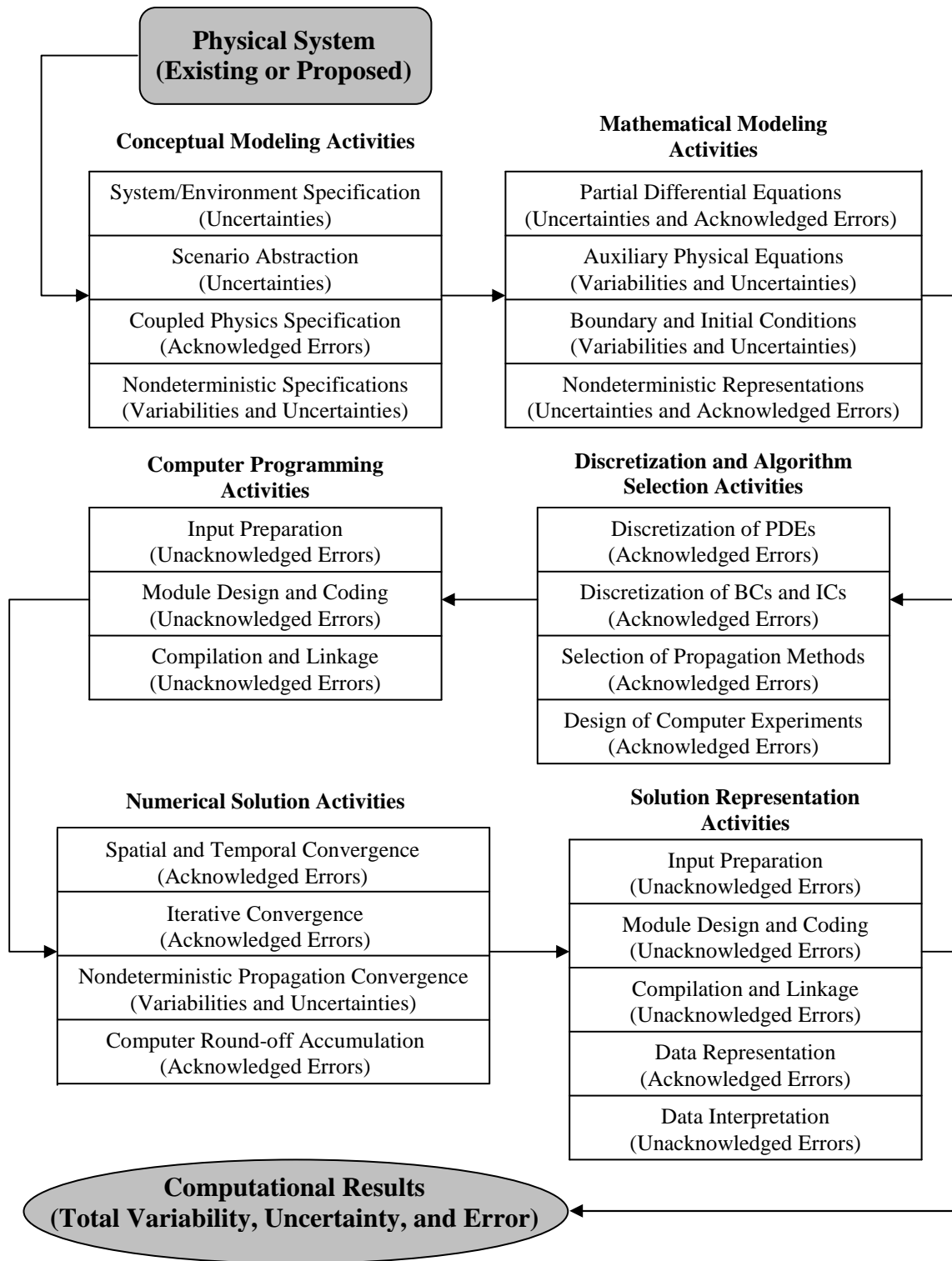


Figure 2-9: Framework for Sources of Uncertainty and Error [74]

The research of design techniques to handle uncertainties that pervades all areas of science and engineering has been an interesting and fundamental research topic for engineers and scientists for a long time. Traditionally, the uncertainty is accounted for in design by the use of scaling parameters such as safety factors, and this kind of method has proved useful by past decades of experiences. However, when new configurations or materials are used, it is difficult to determine a proper value for the scaling factor; besides, the measures of reliability or robustness can not be given [15].

The uncertainty analysis or uncertainty based design has become a crucial technique in many engineering fields such as the aerospace industry. Once it is realized and adopted, the following potential benefits can be achieved [15]:

1. Increase of confidence in analysis or simulation tools;
2. Reduction of design cycle time, cost, and risk;
3. Increase of system performance while meeting the reliability requirements;
4. Increase of robustness of the system;
5. The performance or behavior of the system at off-nominal conditions can be evaluated.

Many advanced methods have been developed for this purpose, such as probabilistic design, fuzzy logic, and interval analysis [15]. The probabilistic design methods are very important and popular means to deal with the pervasive uncertainties, because the difference between model-based prediction and reality caused by uncertainties can be described by probability distributions. There are two developing fields in this area, one is reliability design, and the other is robust design. The detailed task of reliability design is different in different engineering fields. From the viewpoint of

operation, it is to design a system, or component, or device to perform without failure for a specified period of time under specified operating conditions; from the viewpoint of some disciplines such as structural design, it is to prevent catastrophic failures. The robust design, on the other hand, is to obtain less variation of performance or maintain good performance at off-design conditions. One uncertainty classification and corresponding design methods are shown in Figure 2-10.

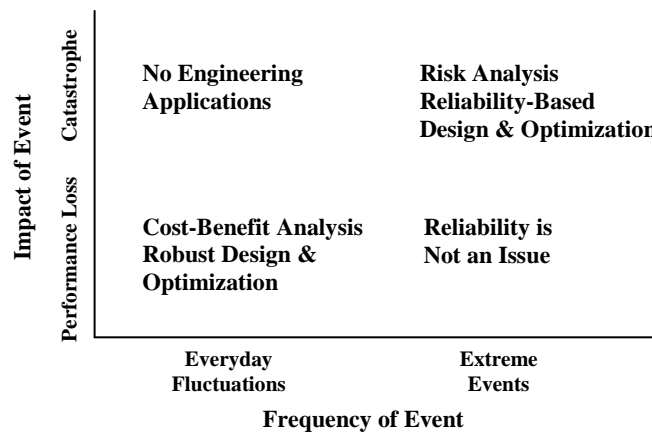


Figure 2-10: Uncertainty Classification and Design Domains [75]

Generally, reliability design deals with extreme events, or the “tails” of a probability distribution, while robust design is interested in the behavior in the zone around a nominal value such as the mean of a probability distribution. This difference is illustrated in Figure 2-11.

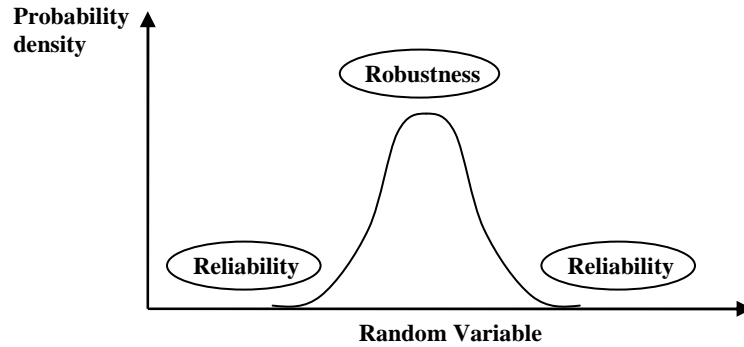


Figure 2-11: Reliability versus Robustness on Probability Density Functions [15]

For complex engineering systems, probabilistic design methods are realized typically using random sampling techniques such as Monte Carlo simulation (MCS) or other statistical sampling methods such as Fast Probability Integration (FPI). This is because for a complex engineering system the real functional relationship between a response and design variables is usually implicit and in general it is very difficult to establish its explicit form, the analytical way of uncertainty propagation by derivative analysis of the functional relationship can not be used.

Monte Carlo simulation is the most popular random sampling technique for complex engineering systems because of its ability to obtain the most accurate probability distribution [76]. However, a large number (in thousands) of complete analyses are required in order to obtain an accurate probability distribution using this technique. For small event probability values, an even larger number (in tens or hundreds of thousands) of analyses are required because the accuracy of Monte Carlo simulation decreases rapidly for lower and lower event probability values. For example, it is a low requirement to require the structural failure rate to be less than 0.01% for an important building in its designed life time. Because this failure probability means 1 failure out of 10,000 scenarios or cases analyzed, more than 10,000 cases are needed in order to obtain an

accurate distribution. The following Equation 2.44 is used to estimate the probability calculation error ε (%) of the Monte Carlo simulation [77]. With this equation the number of analyses needed for $P_f = 0.01\%$ with an error of no greater than 5% is estimated as 80,000.

$$\varepsilon = \sqrt{\frac{200 \cdot (1 - P_f)}{N \cdot P_f}} \quad (2.44)$$

where N is the number of different analyses or simulations required, P_f is the failure probability, and ε is the maximum error (%) of P_f .

Obviously this approach is unaffordable in terms of time if the time-consuming physics-based models such as finite element based analysis are used to do the Monte Carlo simulation, and thus again the fast surrogate models are required.

The probabilistic design methods based on surrogate models and the Monte Carlo simulation have been widely applied to early design stages in engineering fields, such as undersea weapon system design [78], car crashworthiness design [79], and gas turbine blade reliability design [80], to name a few. Significant improvements to the system performance and accuracy of the reliability and robustness assessments have been made by these methods. New characteristics of probability distributions of the system performance have been found, which might not be discovered otherwise, such as in Ref. [80], the distribution of the core temperature of a gas turbine engine is non-normal instead of normal that has been assumed for a long time.

It has to be pointed out that the probabilistic design result also depends on how accurate a surrogate model captures the variations of the responses with respect to the perturbations of the design variables [81]. However, it is more difficult to check this kind

of accuracy than goodness of model fitting, and it is assumed that the more accurate the surrogate models are, the better the ability of the surrogate models is to capture those variations.

2.5.2 Mathematical Foundation of Joint Probability Assessment

The probability and statistics theories of a single random variable are the standard contents of probability and statistics textbooks, and those contents are omitted here.

The general expression for the joint probability of multiple arbitrarily distributed continuous variables is given as

$$P[(X_1, X_2, \dots, X_n) \in A] = \int \dots \int_A f(x_1, x_2, \dots, x_n) dx_1 dx_2 \dots dx_n \quad (2.45)$$

where A is the event space, $f(x_1, x_2, \dots, x_n)$ is the multivariate joint probability density function (PDF). This joint PDF satisfies the following conditions:

1. Positive definite: $0 \leq f(x_1, x_2, \dots, x_n)$

2. Unit integral property: $\int_{\Omega} \dots \int f(x_1, x_2, \dots, x_n) dx_1 dx_2 \dots dx_n = 1$, where Ω is the state

space comprising all possible different events. For continuous random variables, the state space is defined by the intervals of all random variables.

Given a joint PDF, a new concept comes up, i.e. the marginal distribution. A marginal distribution is a univariate distribution function, which can be determined by integrating the joint PDF with respect to the other random variables over the entire state space. As an example, the marginal distribution of the random variable X_1 is given as

$$f_{X_1}(x_1) = \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} f(x_1, x_2, \dots, x_n) dx_2 dx_3 \dots dx_n \quad (2.46)$$

The marginal distribution is important in practice since in most cases one knows the marginal distributions and uses those distributions to construct the joint distribution.

The random variables are mutually independent if and only if the joint distribution function of those variables is equal to the product of corresponding marginal distributions given as

$$f(x_1, x_2, \dots, x_n) = \prod_{i=1}^n f_{X_i}(x_i) \quad (2.47)$$

The conditional probability density function (CPDF) is the (joint) probability density function for some variables given the other variables taken specific values. The most useful CPDF is for a single variable. For example, the CPDF for X_1 is given as

$$f_{X_1|X_2, \dots, X_n}(x_1 | x_2, \dots, x_n) = \frac{f(x_1, x_2, \dots, x_n)}{f_{X_2, \dots, X_n}(x_2, \dots, x_n)} \quad (2.48)$$

where $f_{X_2, \dots, X_n}(x_2, \dots, x_n) = \int_{-\infty}^{+\infty} f(x_1, x_2, \dots, x_n) dx_1$ (using Equation 2.46).

The relationship among the joint, marginal and conditional probability density functions is illustrated by a bivariate case in Figure 2-12.

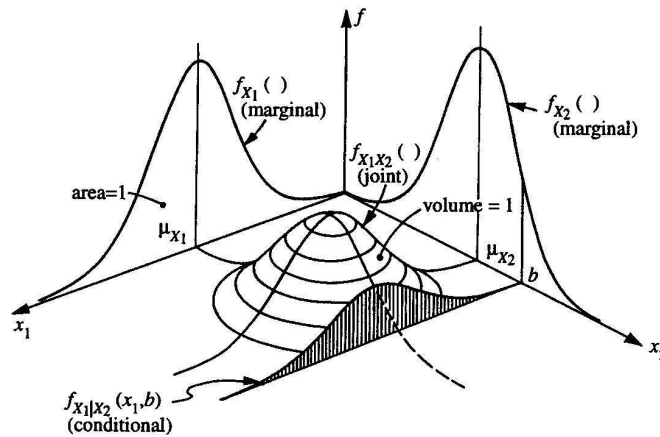


Figure 2-12: Illustrations of Joint, Marginal and Conditional Probability Density Functions [82]

Given the joint density function, the mean (or expected value) and variance of a random variable can be calculated as

$$\mu_{X_1} = E(X_1) = \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} x_1 f(x_1, x_2, \dots, x_n) dx_1 dx_2 \dots dx_n \quad (2.49)$$

$$\text{Var}(X_1) = E[(X_1 - \mu_{X_1})^2] = \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} (x_1 - \mu_{X_1})^2 f(x_1, x_2, \dots, x_n) dx_1 dx_2 \dots dx_n \quad (2.50)$$

The covariance and correlation coefficient between two random variables can be calculated as

$$\text{Cov}(X_i, X_j) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (x_i - \mu_{X_i})(x_j - \mu_{X_j}) f(x_1, x_2, \dots, x_n) dx_i dx_j \quad (2.51)$$

$$\rho_{X_i, X_j} = \frac{\text{Cov}(X_i, X_j)}{\sqrt{\text{Var}(X_i)\text{Var}(X_j)}} \quad (2.52)$$

Let $V = V(x_1, x_2, \dots, x_n)$ a function of the random variables X_1, X_2, \dots, X_n , then usually V is a random variable as well. The mean and variance of V can be calculated with Equations 2.49 and 2.50 by substituting X_1 with V . Further, the cumulative distribution function of V can be calculated by [82]

$$F_V(v) = P(V \leq v) = \int_{R_v} \dots \int f(x_1, x_2, \dots, x_n) dx_1 dx_2 \dots dx_n \quad (2.53)$$

where R_v is the region over which $V(x_1, x_2, \dots, x_n) \leq v$.

2.5.3 Joint Probabilistic Assessment Methods

Many methods have been developed for assessment of the probability of violating either one function of random variables, which is called a limit state function (LSF), or multiple limit state functions. These methods can be classified as two groups: simulation based and analytical. The most popular and most widely used simulation based method is the empirical distribution function (EDF) method based on Monte Carlo simulation. The most popular analytical methods for single LSF are the fast probability integration (FPI) family methods based on the concepts of most probable point (MPP) and LSF [8]. There

are good descriptions of the FPI family methods in Ref. [83], and APPENDIX E provides a concise illustration of the concepts of MPP and LSF.

It is important to assess the joint probability of violating multiple LSF's, because those LSF's are usually correlated instead of mutually independent since those LSF's have some input variables in common. The probabilistic assessment methods for single LSF are skipped here.

2.5.3.1 The EDF Method

The empirical distribution function method first uses empirical data, which are generated with computational simulation, or experimentation, or actual measurements, to generate a sample of the joint PDF or cumulative distribution function based on the simulation counting technique. This sample is fitted to construct an approximate joint PDF or CDF. Then this fitted joint PDF or CDF is used to estimate the joint probability of violating the LSF's.

A joint PDF (sample) is generated using the following simulation counting equation:

$$f(y_1, y_2, \dots, y_m) = \frac{1}{N} \sum_{i=1}^N I(a_{i1}, a_{i2}, \dots, a_{im} = y_1, y_2, \dots, y_m) \quad (2.54)$$

And a joint CDF (sample) is generated using the following equation:

$$F(y_1 \leq a_{i1}, y_2 \leq a_{i2}, \dots, y_m \leq a_{im}) = \frac{1}{N} \sum_{i=1}^N I(y_1 \leq a_{i1}, y_2 \leq a_{i2}, \dots, y_m \leq a_{im}) \quad (2.55)$$

where N is the number of sample points, Y_1, Y_2, \dots, Y_m are response functions of random variables X_1, X_2, \dots, X_n , a_i are pre-specified values for the response functions, and $I(\cdot)$ is the indicator function, giving 1 if the conditions in the parenthesis are all satisfied or 0 otherwise.

Denote $z_j, j=1,\dots,q$ being the q limit state functions of interest. The EDF method is shown in Figure 2-13 by an example for single LSF.

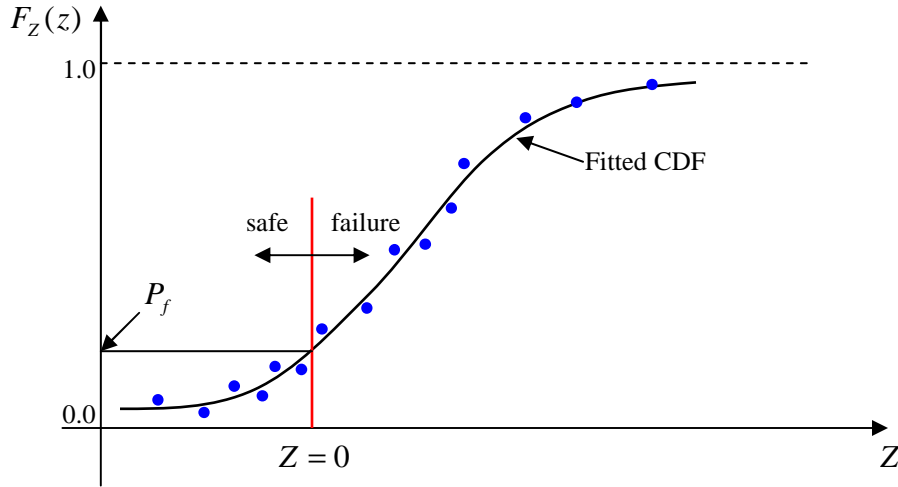


Figure 2-13: Example Using Empirical CDF to Estimate P_f

In practice, however, the joint probability of violating the LSF's ($Z > 0$) is usually directly estimated by the following counting Equation 2.56, instead of being obtained from the fitted joint PDF or CDF.

$$F(z_1 > 0, z_2 > 0, \dots, z_q > 0) = \frac{1}{N} \sum_{i=1}^N I(z_{i1} > 0, z_{i2} > 0, \dots, z_{iq} > 0) \quad (2.56)$$

The most popular computational simulation method is Monte Carlo simulation, which is based on the multi-variate Monte Carlo sampling (see APPENDIX F for details). The basic idea of MCS is simple: first a set of sample points are generated by the multi-variate Monte Carlo sampling method; then the responses of the sample points are obtained. Thus a random sample is obtained or a random process is simulated. The required number of sample points N with an error of no greater than ε can be estimated based on Equation 2.44, which is repeated with here:

$$\varepsilon = \sqrt{\frac{200 \cdot (1 - P_f)}{N \cdot P_f}} \quad (2.44)$$

The Monte Carlo simulation is extremely appealing since it requires the least amount of statistical knowledge and does not need to know the explicit functional relationship between a response function and the random variables. It can be the most accurate approach given enough sample points, and asymptotically converges to an exact answer as the number of sample points approaches to infinity. The last advantage of the Monte Carlo simulation is that it can work with probability distribution functions over finite intervals. This is very important because in reality the interval of a random variable in most cases is finite instead of infinite as in the theory of normal distribution.

2.5.3.2 Analytical Probability Assessment Methods

The basic idea behind the analytical joint probability assessment methods for multiple LSF's is to directly construct the joint probability density function of the responses based on some information of the (random) design variables, or the responses, which are random variables as well.

For the analytical methods that are based on information of responses, such as the Nataf PDF transformation method (NPDF) [82] and Bandte PDF method (BPDF) [8], those methods usually require information of the marginal distributions and/or covariance matrix of the responses. However, in practice, this information is very difficult to obtain, if at all. Therefore, these methods are not practical.

One common problem with the above methods is that the (random) design variables are indirectly linked to the (joint) PDF of the responses and thus these methods are difficult to be used for the design space exploration process. To overcome the problems with the above methods, two methods are developed: multi-response first order second

moment method (MFOSM), and multi-response inverse transformation method (IPDF) [80]. These two methods make use of the partial derivatives of the responses with respect to the (random) design variables while constructing an approximate joint PDF of the responses, and thus are classified as sensitivity-based methods. The partial derivatives are approximated numerically using methods such as finite difference. Those methods are practical because those methods do not require distribution information of the responses and thus are easy to use. In addition, the (random) design variables are now directly incorporated into the joint PDF of the responses and thus these methods can be used for the design process. The main limitations of MFOSM are that it assumes the responses are normally distributed and makes linear approximation to the responses. The main limitation of IPDF is that it requires the inverse functions of the responses with respect to design variables must exist and be unique. This compels this method to use linear approximation functions of the responses since the inverse functions may not be unique otherwise, although theoretically this method does not have to.

2.5.4 Summary of Robust Design and Reliability Design Methods

As mentioned previously, there are two main fields that handle the effects of uncertainties: the reliability design and robust design. In these two fields, the design variables are treated differently as with the objective functions.

The independent variables can be classified into two groups: control variables (CV) and noise variables (NV). The control variables are the ones that are controlled, or specified, or selected by a designer, such as the operating temperature, the shape and dimensions of a component, the properties of a material, etc. The noise variables are the ones that are out of control of a designer while the product is being manufactured or used

in the field up to the end of its lifetime, such as the random material properties, manufacturing tolerances, random loads, the production quantity, utilization hour, economic range, load factor, ambient temperature, etc.

In robust design, variation is considered only for the noise variables and a noise variable is represented by a probabilistic distribution and a range. A control variable, on the other hand, is treated deterministically with a single value over a range (without using a distribution), and its value is adjusted during the design process. Then the variation (such as standard deviation) of a response from a target value is estimated with the distributions and ranges of the noise variables at a given set of values of the control variables.

The objective of robust design is to find the set of values of the control variables that minimizes some responses and the variation of some other responses with respect to the noise variables by adjusting the control variables and making sure that the response does not violate the limit state function(s). Figure 2-14 illustrates the difference between a robust design solution and a deterministic design solution.

There are many realizations of the robust design methods in different areas, such as Ref. [13] and [84] for aircraft multidisciplinary design optimization, Ref. [78] for submarine weapon system design, Ref. [85] for flexible wing design optimization, Ref. [86] for airfoil shape design optimization, to name a few. One such example is shown in Figure 2-15.

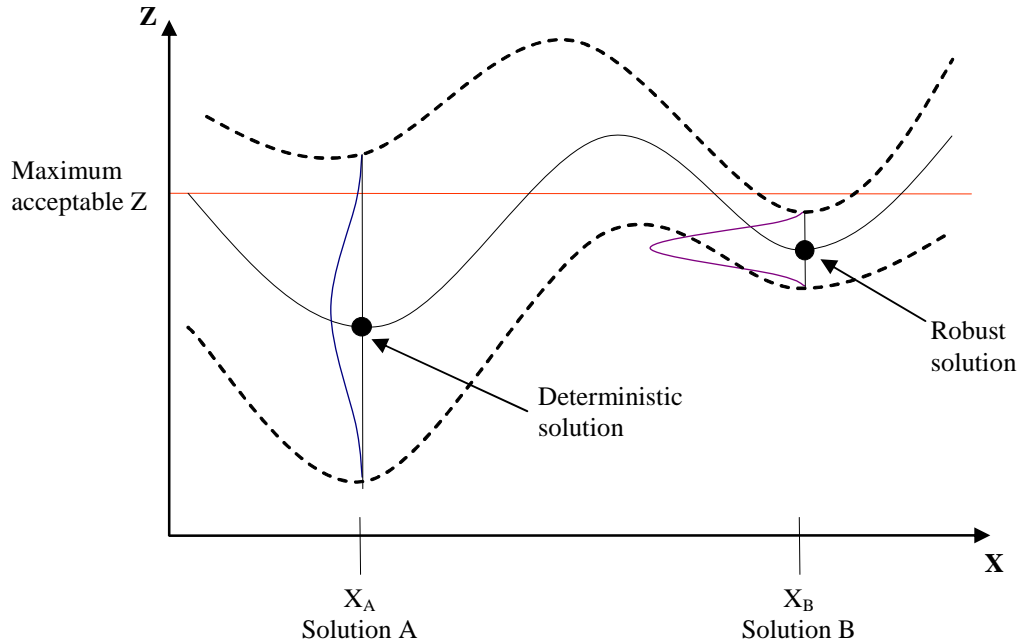


Figure 2-14: Difference between a Robust Design Solution and a Deterministic Design Solution

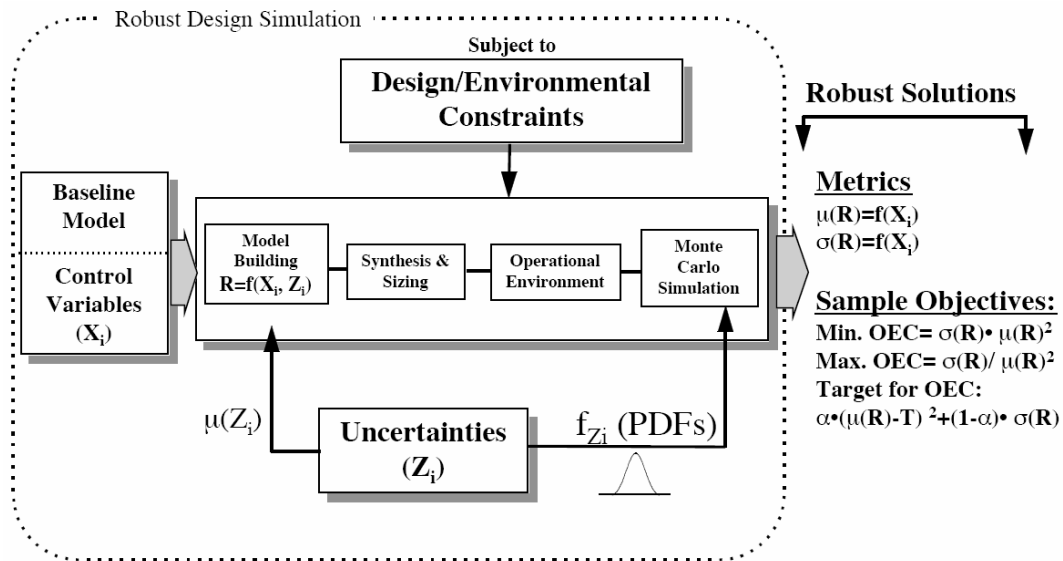


Figure 2-15: Example Implementing Robust Design [7]

In reliability design, variation is considered for both the noise variables and control variables. There are three main approaches to handle the effects of these uncertain variables, i.e. the use of a safety factor, the use of the absolute worst case, and the use of probability [1]. The first two approaches are conventional approaches, and the third one

is the modern reliability design of interest. As in robust design, a noise variable is represented by a distribution and range. However, the representation of a control variable is more complicated than a noise variable. In reliability design, a control variable is represented by a nominal value such as mean value (or expected value), a probabilistic distribution, and a range. Each nominal value may correspond to a specific distribution and a specific range, i.e. for different nominal values the corresponding distributions and ranges can be different. Typically, the nominal values are used to calculate the (average) response that a designer concerns, and adjusted by the designer. The distributions and ranges are used to estimate the probability of violating a criterion or criteria, along with those of the noise variables, using a probability assessment method discussed previously. A probability is estimated for each set of values of the control variables along with the corresponding distributions and ranges.

The objective of reliability design is to find the set of nominal values of the control variables that makes the probability of violating a criterion or criteria less than the target value as well as maximizes or minimizes some performance measures such as \$/RPM and weight, by adjusting the nominal values of the control variables. Figure 2-16 illustrates the difference between a reliability design solution and a deterministic design solution.

Some reliability design methods are a simplified version of the above process. In these simplified methods, a control variable is represented by a distribution and a range, and the designer assigns only one value to a control variable based on these distribution and range, instead of a nominal value and its corresponding distribution and range; and a probability is estimated with the distributions and ranges of all control variables, instead

of for each set of values of the control variables. Sometimes the control variables are even treated as deterministic ones, and the variation is caused only by the noise variables.

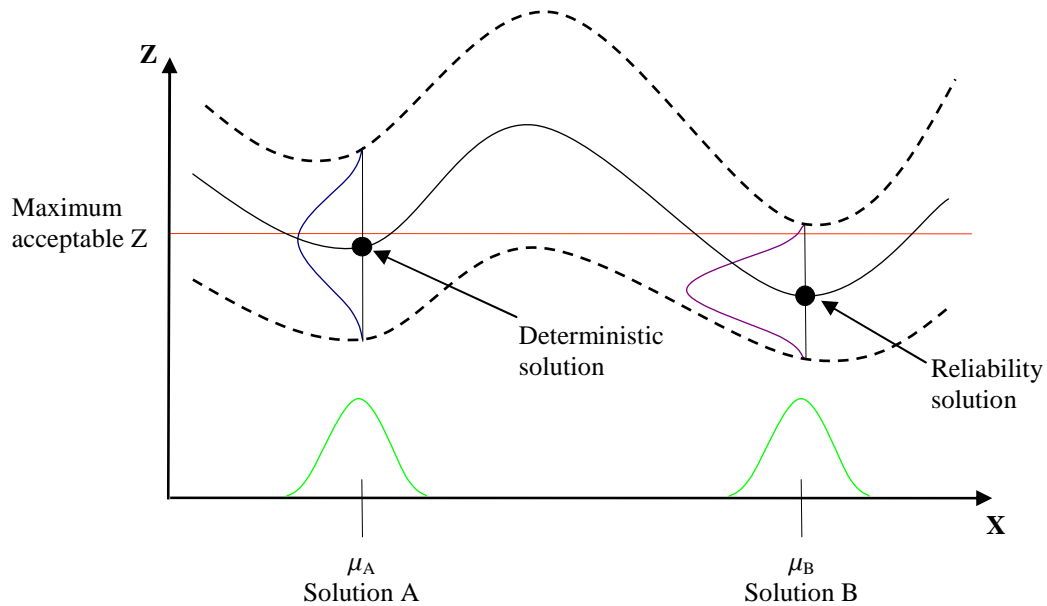


Figure 2-16: Difference between a Reliability Design Solution and a Deterministic Design Solution

A note is that the reliability design methods are usually referred to as probabilistic design methods, which in this case do not include robust design methods. In addition, although the reliability design methods are originally developed to solve the structure reliability problems, those methods are now extended to other problems, such as viability of a product over its life cycle.

There are many realizations of the reliability design methods in different areas, including mechanical systems considering material uncertainties [87], aircraft concept and preliminary design [8], mechanical systems considering manufacturing and operational uncertainties [88], aircraft impact dynamics design optimization [89], to name a few. One such example is shown in Figure 2-17.

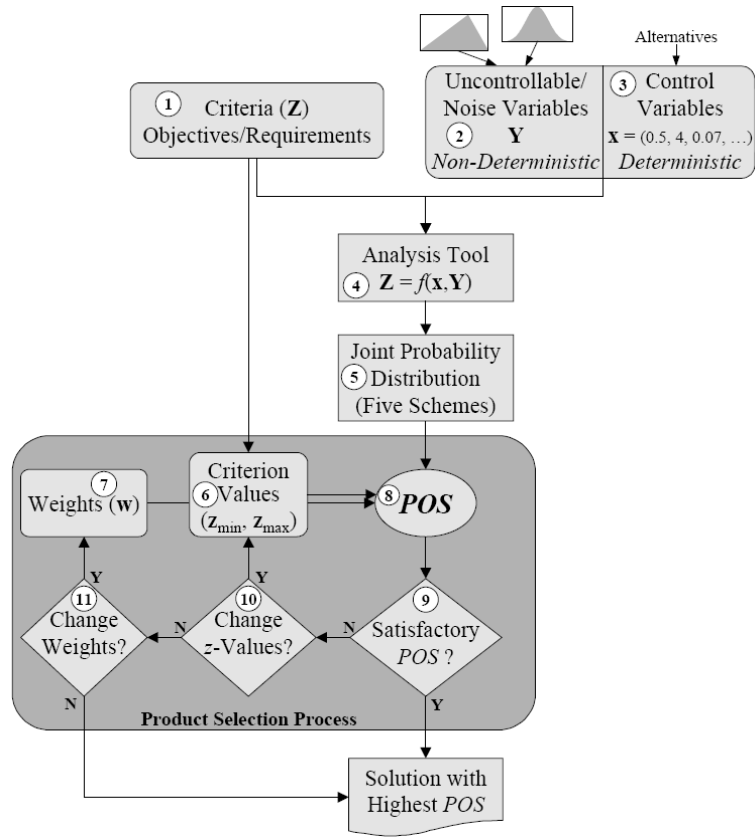


Figure 2-17: Example Implementing the Simplified Reliability Design [8]

2.6 Multi-Objective and Multidisciplinary Optimization Methods

In this section, the basic concepts and methods of multi-objective and multidisciplinary design optimization are summarized.

2.6.1 Multi-Objective Optimization Methods

Often, in a multi-objective optimization problem the criteria are conflicting in such a way that optimization of a single criterion results in poor performance for another criterion. In this case, there is no optimal solution that simultaneously optimizes all the objective functions; instead, the concepts of a Pareto frontier and weak Pareto frontier are employed. Assuming all the objective functions are to be minimized, in the design space the definitions of a PF point (or solution) and a WPF point are given as follows [14]:

- Pareto frontier point: A point X^* is a Pareto frontier point if and only if there does not exist another point X^+ , such that $F(X^+) \leq F(X^*)$, and $F_i(X^+) < F_i(X^*)$ for at least one objective function, where $F_i(X) = f_i(X)$ (see Equation ()).
- Weak Pareto frontier point: A point X^* is a weak Pareto frontier point if and only if there does not exist another point X^+ , such that $F(X^+) < F(X^*)$.

Correspondingly, in the objective space the vector $F(X^*)$ defines a frontier point, either a Pareto frontier point, or a weak Pareto frontier point.

In this research, two kinds of Pareto frontier or weak Pareto frontier are differentiated, i.e. deterministic frontier and probabilistic frontier. The deterministic frontier satisfies all the deterministic constraints, while the probabilistic frontier satisfies all the probabilistic constraints that require first satisfying the deterministic constraints. If there are no special notes, a Pareto or weak Pareto frontier means a probabilistic frontier.

According to the above definitions, a point is a WPF point if there is no another point that improves **all** of the objective functions simultaneously, while a point is a PF point if there is no another point that improves at least one objective function without degrading any other objective functions. Therefore, all PF points are WPF points, but not vice versa. Because of relaxation by definition, the WPF solutions are more useful for practical applications than the PF ones. Figure 2-18 shows the difference between PF and WPF points in the two dimensional objective space. In this figure, the PF points are points b, e, and g, while the WPF points are points a, b, c, d, e, f, g, and h. The Figure 2-19 shows the difference in the three dimensional objective space. It can be seen that the difference is very obvious in this case that the WPF includes more points that form a different section of the edge beside the common section of the PF and WPF points.

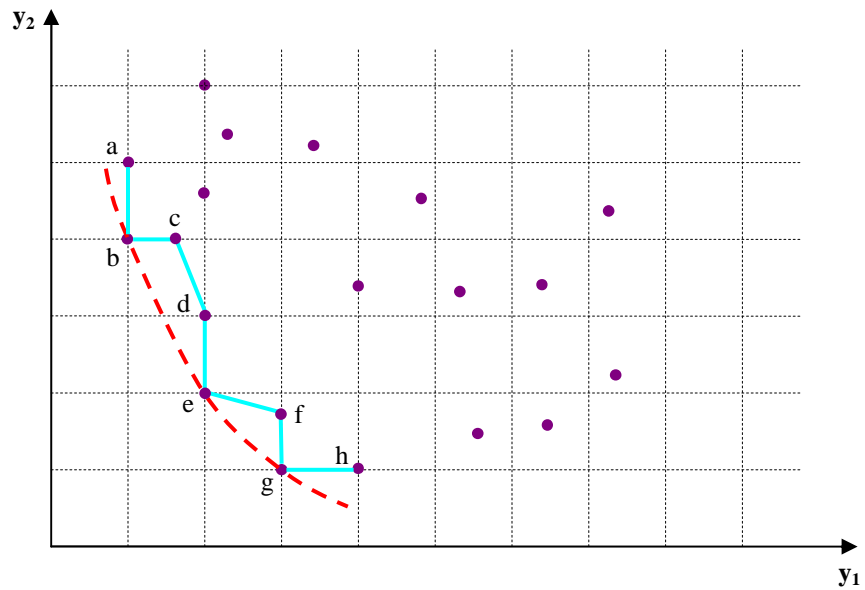


Figure 2-18: Example to Show the Difference between PF and WPF in the 2D Objective Space

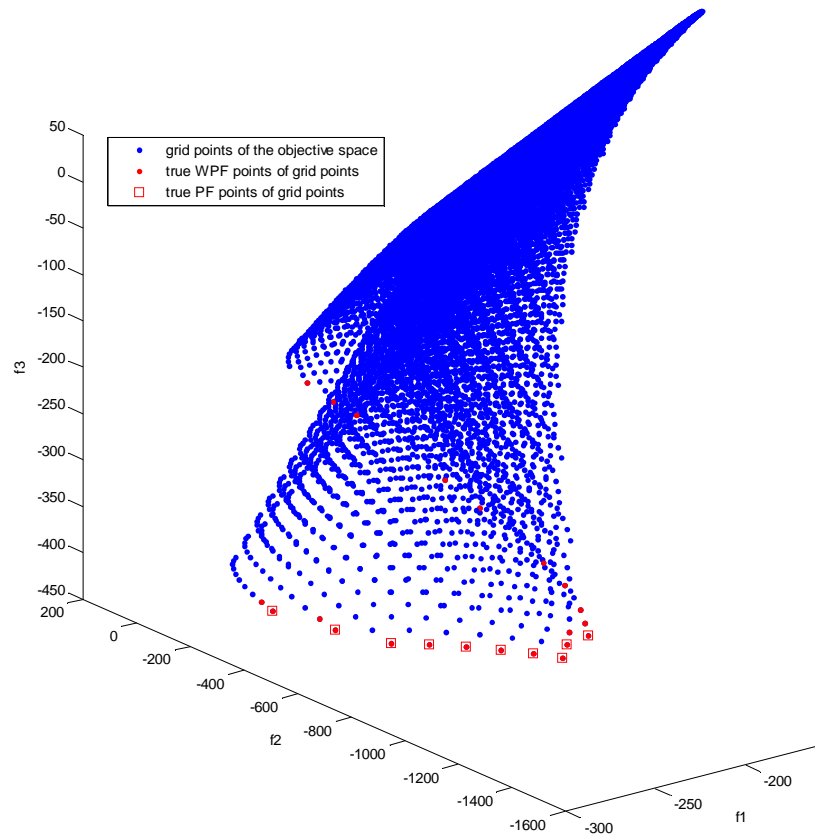


Figure 2-19: Example to Show the difference between PF and WPF in the 3D Objective Space

It can be shown that the methods to obtain PF solutions can easily be modified to obtain WPF solutions. Therefore, no difference is made for the methods to obtain PF or WPF solutions. Many methods have been developed to find the optimal solution(s) of a multi-objective optimization problem in the sense of PF or WPF. Reference [14] provides a good survey of these methods.

These methods can be divided into two groups, one that needs articulation of preferences, and one that does not. The first group usually uses a preference parameter, such as a weight for an objective function to show the preferences of the decision-maker either explicitly or implicitly. With the help of the preference parameter, the objective functions can be combined to form a single (objective) utility function, such as the most common one, weighted overall evaluation criterion method, or a single utility function without direct information of the objective functions, such as the Goal Attainment method [90]. After forming the single utility function, a single-objective optimizer is used to find a multi-objective optimal solution. By varying the preference parameter systematically, a set of MOO solutions, i.e. the PF or WPF solutions, can be found. Usually the constraints are treated directly by the single-objective optimizer, i.e. without using a penalty approach that further increases the complexity of the utility function.

The standard form of the Goal Attainment optimization problem is as follows:

$$\begin{aligned}
 & \text{Minimize: } \lambda \\
 & \text{Subject to:} \tag{2.57} \\
 & F_i(X) - w_i \lambda \leq b_i \quad i = 1, e \\
 & w_i \geq 0
 \end{aligned}$$

where

λ	a scalar variable
w_i	weights
b_i	goals for the objective functions

Figure 2-20 shows the basic idea of the Goal Attainment method, in which W and B are the vectors of weights and objective goals, respectively. From this figure, the vector of the objective goals is better to be outside of the objective space F , and the Pareto frontier of the objective space can be either convex or concave.

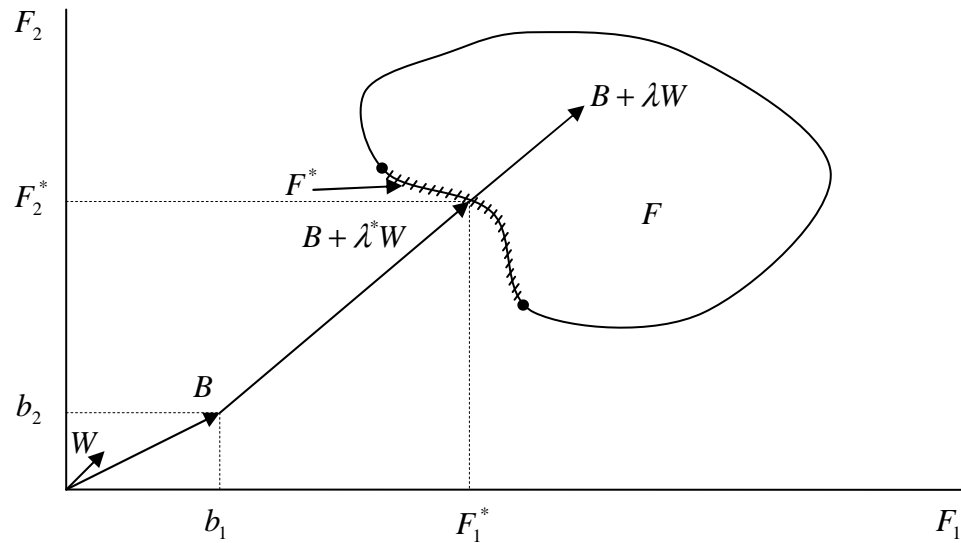


Figure 2-20: Illustration of the Basic Idea of the Goal Attainment Method (adapted from [90])

A very important conclusion that one can draw from this figure is that the weights vector W is not used to put different weights or preferences on different objectives; instead, it is just a means to define the search direction. Therefore, the Goal Attainment method is actually a non-preference method in the author's opinion, and it should not be classified as a method with preferences.

The second group does not need the preference information to find the PF or WPF solutions. In contrast to the first group of methods treating the objective functions indirectly and finding one PF or WPF solution at a time, this group treats the objective functions directly without forming a single utility function and provides a set of PF or WPF solutions as a whole. The most popular method of this group is the Genetic Multi-objective Algorithm. One disadvantage of the Genetic Multi-objective Algorithm is that the constraints are not addressed directly and usually a penalty approach has to be used. In this research, a new Monte Carlo simulation based method is formed to find the WPF solutions.

These methods can also be divided into another two groups, one that finds a single PF or WPF solution at a time, and one that directly generates a set of solutions. An example of the first group is the Goal Attainment method discussed above. Special procedures are required to find a set of PF solutions with the first group methods, since it is desirable that such a method has practical attributes: a) it should generate evenly distributed PF points in the objective space; b) it should explore the entire objective space and not neglect any region [91]. To generate a set of evenly distributed PF points of the whole objective space, one approach is to systematically change the parameters in a single solution MOO method, such as the weights in the OEC method. However, this approach does not always result in an even distribution of PF points even though the weights are evenly varied [91]. Another approach is to directly generate such a set of PF points, such as the modified Normal Constraint (NC) method [92], and the new method that will be provided later in this research. The main reason not to use the modified NC method and instead to develop a new method in this research is that the modified NC

method needs non-trivial optimization effort to generate initial search points, whereas the new method does not.

2.6.2 Multidisciplinary Optimization Methods

As mentioned previously, a multidisciplinary design optimization problem features coupling variables, and these variables and the design constraints make the design very complicated. First, design freedom is reduced and disjointed consistent design zones result in the system level design space. Second, those methods entail equality constraints for the coupling variables in the multidisciplinary analysis process and thus require many iterations of multidisciplinary analysis in order to find every single consistent design point. Third, special solving procedures are required to untangle or decompose the complex interactions introduced by coupling variables and to find consistent design points. Last, the design constraints further reduce the design freedom and entail more effort to find final feasible design solutions.

Figure 2-21 shows the design structure matrix (DSM) of a transportation aircraft multidisciplinary design problem (see APPENDIX H) that will be solved later. The boxes D, A, W, and P represent the disciplinary analysis of zero-lift drag contributing analysis (CA), aerodynamics CA, weights CA, and performance CA, respectively. In this DSM, there are two coupling variables, i.e. V_{br} and $W_{landing}$. V_{br} is one input of the D CA and also one output of the A CA; $W_{landing}$ is one input of the A CA and one output of the W CA. The design variables are b , l , S , W_{to} , and T_i .

Depending on availability of optimization ability in the disciplinary contributing analyses, different approaches are developed to find the optimal solutions from the consistent design zones. If there is not optimization ability in the disciplinary CA's, the

common approaches are optimization with (relaxed) Fixed Point Iteration method (FPI) and optimizer based decomposition method (OBD) [93]; if all the disciplinary optimization ability is turned off and combined into a single system-level optimizer, the approach used is called All-at-Once method (AAO) [94]; if the disciplinary optimization ability is to be kept (only part of it in fact), multi-level MDO methods are used, such as the Collaborative Optimization method (CO) [18], Modified Collaborative Optimization method (MCO) [95], and Bi-Level Integrated System Synthesis method (BLISS) [17].

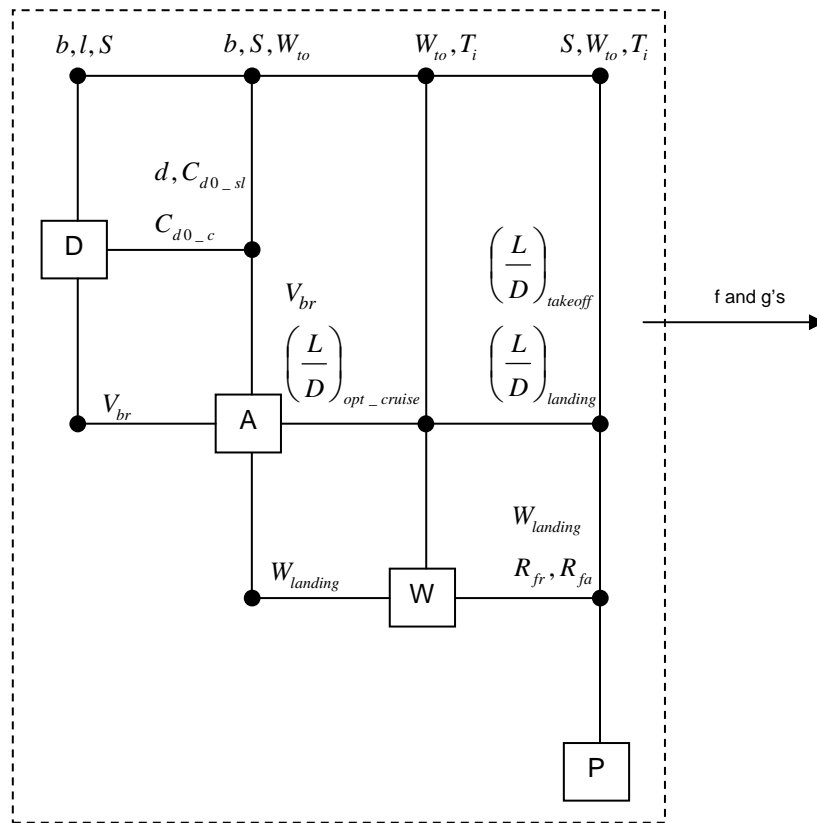


Figure 2-21: Example of the DSM of a Multidisciplinary Aircraft Design Problem

As discussed previously, the new framework to solve the JPMOMDO problem that represents a realistic conceptual design problem needs to be a loosely coupled or completely decoupled architecture in order to reduce or eliminate the nesting loops. Of all

the MDO approaches above, OBD can completely decouple the problem and the resulted alternative problem is easier to understand than the others. This approach should be considered by the new framework.

3 RESEARCH QUESTIONS AND HYPOTHESES

In Chapter 1, a need is established to formulate a new framework for realistic conceptual design problems of design alternative generation and selection. In Chapter 2, the state-of-the-art enabling techniques for the new framework were reviewed, which included methods of surrogate modeling, joint probabilistic assessment, probabilistic design, multi-objective optimization, Pareto frontier finding, and multidisciplinary optimization. Based on the desired elements for the new framework discussed in Chapter 1 and state-of-the-art enabling techniques and methods reviewed in Chapter 2, research questions are raised for the formation of the new framework and three hypotheses are proposed.

3.1 About Formulation of the New Framework

One idea proposed about the formulation of the new framework is to combine some of the presented methods together. Since each of those methods is good at solving a specific problem, a hybrid approach might be able to handle a more general set/category of problems.

However, when making use of the advantages of those methods, the disadvantages also need to be overcome. For example, each of the methods available for consideration of MDO, JPA, MOO, and PF implies considerable computational load and run time since those methods usually find a solution by iteration. For a MOO method, it is difficult to find a PF point in the objective space, and it is much more difficult to find a corresponding PF solution in the design space since usually the design space is not

involved when finding a PF point in the objective space; a probabilistic design method faces similarly serious difficulty when finding a corresponding solution in the design space.

Therefore, when combining the methods of MDO, JPA, MOO, and PF finding together, the nesting loop approach can **not** be adopted, which gives a loop to each of those methods, because this approach will entail unacceptable computational load and time. Those methods need to be combined in a decoupled way. In addition to this decoupling requirement, the weak Pareto frontier points in the objective space and corresponding design solutions in the design space should be found at the same time, in order to avoid the difficulties that the current methods are suffering.

The observation on current practice shows that the combination of the above four kinds of methods has not been achieved yet, although some of those methods are combined, such as the TIES, RDS, component reliability assessment method [80] combining MDO and joint probability assessment (JPA); the fuzzy Pareto Frontier method [96] combining MDO, MOO, and finding PF; and the aero-propulsion component design method [97] combining MDO, MOO, and a separate probabilistic assessment.

One obvious reason for this combination not being done is that no existing approaches can combine those four kinds of methods together but in a decoupled way, and also enable finding the weak Pareto frontier points in the objective space and corresponding design solutions in the design space at the same time, although the above examples show some approaches can combine some, not all, of those methods. Another possible reason is that those methods may not work well with each other since those

methods are developed separately. For example, the MDO methods usually are developed to solve a single objective problem and thus do not work well with MOO methods, and vice versa; the analytical probabilistic assessment methods do not perform well with complex multidisciplinary models as shown by experiences of the author and many other people; the Monte Carlo probabilistic assessment method can work well with complex models, but the computational load may be huge if it is directly combined with a MDO or MOO method.

As discussed in Chapter 1, accurate approximation methods are desired for the new framework in order to perform rapid assessment and make use of accurate or high fidelity knowledge. There are more reasons for accurate approximation methods. Since the conceptual design decisions have very important effects on final performance, quality, and 70 to 80 percent of the cost [1], and the probabilistic assessment and design results are sensitive to the accuracy of the surrogate models, these surrogate models must be accurate enough [9, 15, 81] in order to obtain trustworthy probabilities of the PC's and subsequent WPF.

Considering the fact that different high fidelity but time consuming tools are used to create the training samples for construction of surrogate models when a design is revolutionary, there should be a surrogate modeling method to be accurate for many types of problems with a small training sample. This method does not need to be the most accurate one all the time, but it really needs to have broad adaptability with a small training sample. Although one can try several surrogate-modeling methods at the same time and pick the best one, it is possible that none of these methods are good. Therefore,

to avoid this situation, there is a need to have a surrogate-modeling method that is known to be accurate for many types of problems with a small training sample.

The previous comparison of major surrogate modeling methods in Chapter 2 shows that all those methods have advantages and disadvantages, and there is no single method that is superior to the others in all circumstances. In other words, none of those methods has the desired broad adaptability, needless to say high accuracy with a small training sample. Instead of creating a brand new method, it is hoped some existing methods can be combined to keep the advantages and overcome the disadvantages of those methods, so that the resulting hybrid method is accurate for many types of problems with a small training sample.

In fact, hybrid surrogate-modeling is not a new idea. In Ref. [98], RSM and ANN has been combined together to achieve better approximation capability. In this combination, RSM is used to capture the global tendency, and ANN is used to capture (local) high non-linear behavior. However, because of the disadvantages of ANN, this hybrid surrogate-modeling method has not been widely accepted, although RSM itself has been widely applied to many engineering problems.

Inspired by the hybrid method of RSM and ANN, RSM and SVR are considered to construct a new hybrid method. Second order RSM has been accepted by engineers and has been widely applied to various engineering problems. It is very easy to use, very transparent, very fast, and very accurate for low nonlinear problems. With the form of polynomial functions obtained by second order RSM, the contributions of different design variables and the interaction terms of those variables can be easily identified. This kind of information provides more insights into the system behavior and can be used to

improve reliability and robust designs. Therefore, second order RSM is strongly recommended that it should be executed first to see if a reasonable fit can be obtained [60]. However, second order RSM is not accurate for high nonlinear problems, or in these cases only accurate in a small neighborhood because of its mathematical foundation of Taylor series expansion. For multi-objective optimization, the region of interest will rarely be reduced to a small neighborhood by optimization [84]. It suffers the problem of “curse of dimensionality”. Improvement to the accuracy of the response surface models goes slowly, if at all, with increase of the size of the sampling data, because if the order of the polynomial is selected, the number of coefficients is known; then if the sample size is larger than the number of coefficients, the extra data will help little with the accuracy.

SVR, although it is a new method, has been shown to be robust, accurate with good computational efficiency, and have good functional explicitness comparable to that of second order RSM as discussed in Chapter 2. SVR is accurate for many high nonlinear engineering problems [35], and does not suffer the problem of “curse of dimensionality” because of its solid theoretical foundations using the SRM principle as the risk minimization principle.

The other three surrogate modeling methods discussed, i.e. Kriging, Gaussian Process, and Neural Network, are not selected for several reasons. Kriging and Gaussian Process are particular surrogate-modeling methods, making assumptions directly or implicitly about the distribution of the error in Equation 2.14, i.e. Kriging needs to select proper correlation functions and Gaussian Process needs to assume that the error distributions are independent normal distributions. When the error distributions assumed are quite different from the real ones, the accuracy will be low. Those methods also suffer

the problem of “curse of dimensionality” inherited from the particular or parametric inference method. And for Kriging only, it has the problem of low construction speed. For Neural Network, although it is a general surrogate-modeling method and does not have the problem of “curse of dimensionality”, it has problems with the training process, i.e. it is difficult to select a proper training optimization algorithm for all kinds of problems. An improper training optimization algorithm will result in a local minimum solution, or overfitting because of the ERM principle as the risk minimization principle. The last reason is that the accuracy level of those methods is not better than that of SVR according to the comparison provided in Chapter 2.

Therefore, it is believed by this author that a hybrid surrogate-modeling method with the combination of second order RSM and SVR (RSSVR) will be accurate for many types of problems with a small training sample. This new method is needed by the engineering practice and will make improvements to designs with surrogate models.

The second order RSM will be referred to as RSM hereafter.

A good SM for a given problem should be both accurate and simple. In other words, if the accuracy level of two SM’s constructed by two different SM methods is similar, the simpler one should be used; or if the complexity level of two SM methods is similar, the more accurate SM should be used. Since the model accuracy can be measured by model fitting error and model predicting error, this requires a good model selection advisor to balance the model fitting error, model predicting error, and model complexity.

Unfortunately, all existing model assessment methods either do not use all of model fitting error, model predicting error, and model complexity, or have other shortcomings. For example, the coefficient of determination (R^2) measures the model fitting error or

goodness-of-fit, it almost invariably increases and never decreases with the number of parameters, and thus can not be used as a model selection criterion [65]; instead, the adjusted R^2 corrects this problem with an adjustment to the number of parameters, and is widely used as a model selection criterion [68]. The hypothesis testing procedure needs subjective judgment on the levels of significance, therefore there are ambiguities in this method, and it can not be used as a model selection method [63]. There are also other criteria developed during the past based on the concepts similar to model fitting error, such as C_p criterion and S_p criterion, but those criteria were not widely adopted. Cross validation and bootstrap methods estimate the model predicting error. When those methods are used for model selection, one additional disadvantage is that those methods are time consuming. The information criteria AIC and BIC balance model fitting error and model complexity, and this is a great improvement. Although AIC and BIC have been successfully used to select the best model for many surrogate-modeling methods, it has been reported that those criteria have difficulties to select models for the neural network method, i.e. those criteria fail to reliably select the best model [68, 99]. The main reason is that there are typically a large number of parameters to be estimated in an ANN model such that an ANN model can have very low model fitting error but high model predicting error, i.e. the problem of overfitting. This observation confirms that it is not enough to include just the model fitting error and model complexity, instead, inclusion of the model predicting error is also needed. Therefore, a new model selection advisor needs to be created that make use of and balance model fitting error, model predicting error, and model complexity.

Additionally, since the current methods to estimate the model predicting error is time consuming, such as cross validation and bootstrap, a new method may need to be formulated to estimate the model predicting error at a much faster speed than cross validation and bootstrap.

3.2 Research Questions

Based on the considerations of the new framework, three top-level questions are first asked, and then detailed questions are listed in order to develop this framework into a specific method.

Research Question A: Since RSM is good at capturing the global tendency and SVR is good at capturing (local) high non-linear behavior, is it possible to combine these two methods to make a new hybrid method that can be accurate for many types of problems with a small training sample?

Various factors affect the success of a surrogate-modeling method. These factors include the nonlinearity of the model, the dimension or number of the design variables, data sampling techniques, size of the sampling data, and pre-specified parameter settings of the surrogate-modeling method. In order to form a new hybrid surrogate-modeling method of RSM and SVR, the following questions are asked:

1. Although it has many advantages and good characteristics, can SVR be used directly in engineering problems like RSM has been impressively demonstrated in the past? Or what means should be taken to make it suitable?

2. How can RSM and SVR be combined to form a new hybrid surrogate-modeling method that is accurate for many types of problems with a small training sample?

3. Using the previous five criteria for comparison in Chapter 2, is this hybrid method of RSM and SVR better than RSM or SVR for engineering problems? Or under what situation is it better?

4. Is it possible to quantify the five criteria, such that the above comparison in Question 3 can be reliably made?

5. Is it possible to create and formulate a process for which all pre-specified parameters of SVR can be determined automatically such that this hybrid surrogate-modeling method is as simple to use as RSM?

6. Is there a kernel function for SVR that can work well for all engineering problems? If not, how to select a kernel function for different problems?

7. What is the best data sampling technique for this hybrid surrogate-modeling method?

The Question 6 above is important to make this method practical to the average engineers.

Research Question B: Since none of the current surrogate model selection methods balances model accuracy and complexity, where model accuracy is measured by model fitting error or model predicting error, is it possible to make a new method that will achieve this kind of balance?

There are many quantitative measures of model accuracy and complexity. Selecting proper quantitative measures of model accuracy and complexity and properly combining these two kinds of measures together are the keys of the selection advisor of the surrogate-modeling methods. In order to form a new selection advisor, the following questions are asked:

8. What quantitative measures of model accuracy and complexity are appropriate for the purpose of selection of surrogate-modeling methods?

9. What is the proper way to combine the measures of model accuracy and complexity together so that a balance is achieved between these two kinds of measures?

10. When the accuracy is at the same level, can the selection criterion select the surrogate model constructed with the simpler surrogate-modeling method?

Research Question C: Since current multidisciplinary optimization methods, multi-objective optimization methods, and joint probability assessment methods are developed in parallel, is it possible to form a new framework to combine those methods all together but in a decoupled way to solve a joint probabilistic constraint, multi-objective, multidisciplinary optimization problem, and at the same time find the WPF solutions?

The feasibility of a new framework depends on several factors, such as how to find consistent designs, how to find WPF points, how to relax the thresholds in the PC's because of the errors introduced by the surrogate models, et al. In order to form a new framework for determination of the WPF design solutions under probabilistic constraints, the following questions are asked:

11. At which level is the surrogate model constructed, i.e. at disciplinary or system level?

12. How can a consistent design solution be found with this framework?

13. Can the optimal consistent design solutions of the single-objective optimization problems with deterministic constraints be found, or near solutions be found?

14. How can the WPF of each disjointed consistent design zone be found?

15. How can the number of search starting points be selected such that an appropriate number of WPF points can be found?

16. How can evenly distributed WPF points be found for practical usefulness?

17. Because of the errors introduced by the surrogate models, how can the thresholds in the PC's be relaxed such that trustable probabilities can be obtained?

18. What is the best scheme for this new framework in terms of ability to find WPF solutions and computational time?

3.2 Hypotheses

Based on the considerations about the formulation of the new framework and the research questions, the hypotheses of this research are proposed as follows:

Hypothesis A: A hybrid surrogate-modeling method based on a combination of RSM and SVR is not only feasible for complex physics-based models, but also makes improvement over either RSM or SVR where either one of RSM and SVR can not obtain satisfactory results, and can obtain high accuracy for many types of problems with a small training sample

The assessment criteria to support the above hypothesis A are as follows:

1. Accuracy for different complexity (order of nonlinearity) of test problems, under different sample sizes (scale of the sample data);
2. Robustness in terms of variance of error values for different samples generated by different sampling methods;
3. Efficiency in terms of time used for surrogate model construction and new predictions;
4. Transparency in terms of function relationship and factor contributions;

5. Simplicity in terms of the number of parameters needed to be specified by a user;
6. Vulnerability to the problem of “curse of dimensionality”.

Hypothesis B: A surrogate model selection method based on a modified information criterion can select the best surrogate-modeling method for a given problem in terms of balance between accuracy and complexity, where accuracy is measured by both the model fitting error and model predicting error. Specifically, in this research the candidate surrogate-modeling methods are the second order RSM, SVR, and the hybrid method of these two.

The assessment criteria to support the above hypothesis B are as follows:

1. A quantitative measure or measures of model accuracy for comparison of surrogate models constructed by different surrogate-modeling methods;
2. A measure or measures of model complexity for comparison of surrogate models constructed by different surrogate-modeling methods;
3. A combined measure of model accuracy and complexity that achieves a balance between accuracy and complexity for comparison of surrogate models constructed by different surrogate-modeling methods.

One thought is that, it may be very difficult, if at all, to develop a method advisor that can select the best one from all known surrogate-modeling methods. For this reason, in this research the method advisor to be developed is only required to select the best one from RSM, SVR, and the hybrid method of RSM and SVR to be formulated later in this research. Whether it can be extended to select from more methods can be future work.

Hypothesis C: A Monte Carlo simulation based method can be used not only to obtain probabilities of satisfying the PC’s, but also to find the weak Pareto frontier in the

objective space that jointly satisfies the PC requirements and the compatibility constraints for the coupling variables among the disciplinary analyses. Thus a new framework that is based on the Monte Carlo simulation method can determine in a decoupled way the WPF design solutions under probabilistic constraints for a multi-objective, multidisciplinary design optimization problem.

The assessment criteria to support the above hypothesis C are as follows:

1. Ability to find consistent design solutions;
2. Ability to find the optimal consistent design solutions for single-objective optimization problems with deterministic constraints, or solutions very close to these optimal single-objective ones;
3. Ability to find the weak Pareto frontier;
4. Ability to find an appropriate number of weak Pareto frontier points;
5. Ability to find evenly distributed weak Pareto frontier points.

4 FRAMEWORK FORMULATION

Now that the relevant background literature has been reviewed, a new Monte Carlo simulation based framework is devised to determine the WPF solutions under PC's for multi-objective and multidisciplinary design optimization problems for design alternative generation and selection.

This framework starts with constructing fast and accurate surrogate models of different disciplinary analyses in order to reduce the computational time and expense to a manageable level and obtain trustworthy probabilities of the PC's and the WPF. The surrogate modeling methods are limited in this research to RSM, SVR, and a new hybrid method that consists of the second order RSM and SVR. The parameters of SVR to be pre-specified are selected using practical methods and a new modified information criterion that makes use of model fitting error, predicting error, and model complexity information. The best surrogate modeling method for a given problem is also selected using this modified information criterion. Then a new neighborhood search method based on Monte Carlo simulation is used to find valid designs that are consistent for the coupling variables featured in a multidisciplinary design problem and satisfy all the deterministic constraints. Two schemes have been developed. One scheme finds the WPF by finding a large enough number of valid design solutions such that some WPF solutions are included in those valid solutions. Another scheme finds the WPF by directly finding the deterministic WPF of each consistent design zone that is made up of consistent design solutions. Then the probabilities of the PC's are estimated, and the WPF and corresponding design solutions are found.

4.1 Surrogate Modeling and Model Selection

This section introduces space filling sampling methods used in this research, some considerations about surrogate modeling, the random cross validation method for model predicting error, the modified information criteria, the hybrid surrogate-modeling method RSSVR, and last the flowchart of the new hybrid surrogate-modeling method and model selection advisor.

4.1.1 The Space Filling Sampling Methods

Latin Hypercube sampling, Hammersley sequence sampling, and Monte Carlo Sample are selected for this research for three reasons. The first reason is that the user can freely decide the number of sample points. The second is that the uniformity and randomness of the sampling points are satisfactory. The last but not the least is that the sampling points can be generated very fast.

Comparing the Latin hypercube sampling and Hammersley sequence sampling methods, the latter has two advantages. One advantage is that the correlation among the design variables of the sampling points is very low, which helps generate surrogate models with high predicting accuracy. The Latin hypercube sampling method can not guarantee low correlation. The other advantage is that the generation of sampling points is repeatable because Hammersley sampling does not use a random number generator, which helps comparison of results and data management, e.g. there is not need to save the sampling points and instead the sampling points can be generated whenever needed.

Based on the above observation, the Hammersley sequence sampling method is used to generate training sampling points for construction of surrogate models, and the

Latin hypercube sampling method is used to generate sampling points for assessment of the model predicting error.

4.1.2 The Ranges of the Design Variables

The construction of accurate surrogate models requires careful selection of ranges of the design variables. These ranges form the design space. The infeasible regions in the selected design space will result in failed cases or outliers. Although the failed cases and outliers can be identified and excluded, too many failed cases and outliers will decrease the fitting accuracy of the surrogate model over the given design space. This problem makes the prediction by this surrogate model doubtful; seemingly good predictions can be obtained for the points in the infeasible regions. In such cases, the design space has to be changed by trial and error to avoid most, if not all, of the infeasible regions.

In probabilistic design, the designers are handling the nominal values (such as mean values, and the most probable value) of the design variables, and thus the design space is the ranges of the nominal values of the design variables. Therefore before construction of surrogate models for probabilistic design, there are two kinds of design spaces to be differentiated. The first design space is the design space for the probabilistic designer, i.e. the ranges of the nominal values of the design variables. The second design space is the extended design space (EDS) over which the surrogate models are constructed. Since the design variables are randomly distributed about every nominal value, the extended design space must be larger than the design space to accommodate the distributions of the design variables. Figure 4-1 illustrates the concept of extended design space of a two-variable design problem. As a rule of thumb, for example, for a normally distributed variable, the lower limit of this variable in the extended design space should be at least 3σ less than

that in the design space, while the upper limit should be at least 3σ greater than that in the design space.

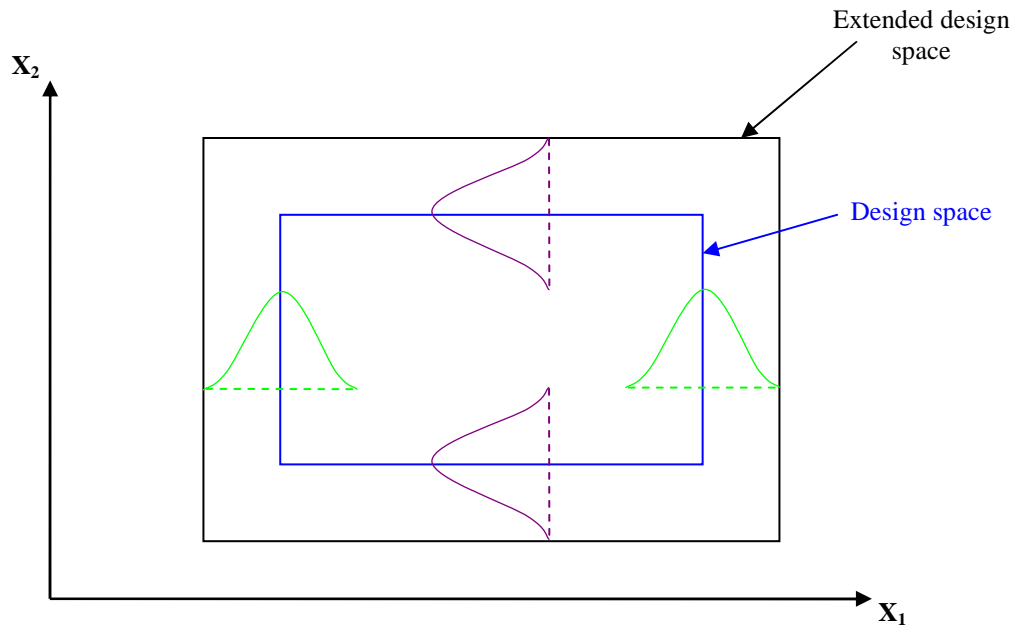


Figure 4-1: Illustration of the Difference between the Design Space and Extended Design Space

With surrogate models constructed over the extended design space, the designers of probabilistic design problems can reduce most of the extrapolation and obtain more accurate results. The extended design space also should be adjusted by trial and error to avoid infeasible regions.

4.1.3 Normalization of Values of Both Design Variables and Responses

It is a common practice to normalize the values of design variables for several reasons. First reason is to avoid the problem of deemphasizing the small-valued variables by the large-valued ones. In an engineering problem usually there are small-valued variables and large-valued ones, for example, a drag coefficient is less than 0.1, while the wing span can be at the magnitude of 100 ft for a transport aircraft. Without normalization, the coefficients of the small-valued variables in the surrogate model can

be very small so that these design variables have no effects on the fitted response. The second reason is that the coexistence of small and large values may make it difficult to inverse a matrix, an operation often needed by a surrogate-modeling method. The last reason is that large values can cause numerical problem if exponential functions are used in the surrogate-modeling methods. One example is the GRBF in the SVR.

In this research, the response values are also normalized. The main reason is to establish a standard process to select the three pre-specified parameters of SVR: the regularization factor C , the deviation ε , and the parameter σ of the GRBF kernel. Without normalization of the response values, it is very hard to determine the criteria for selecting these three parameters because the criteria should change with different magnitudes of response values.

The values of the design variables will be normalized to $[0, 1]$, and the values of responses will be normalized to $[0, 100]$. The following equations are used for normalization:

$$\text{Design variables: } \bar{X} = (X - X^{\text{EL}})/(X^{\text{EU}} - X^{\text{EL}}) \quad (4.1)$$

$$\text{Responses: } \bar{Y} = 100(Y - Y_{\min})/(Y_{\max} - Y_{\min})$$

where X^{EL} and X^{EU} are the lower limit and upper limit of the extended design space, respectively; and Y_{\min} and Y_{\max} are the minimum and maximum values of the response of the training sample, respectively.

To denormalize the responses and RMSE's, use the following equations:

$$\text{Responses: } Y = Y_{\min} + \frac{\bar{Y}(Y_{\max} - Y_{\min})}{100} \quad (4.2)$$

$$\text{RMSE: } \overline{\text{RMSE}} = \frac{\overline{\text{RMSE}}(Y_{\max} - Y_{\min})}{100} \quad (4.3)$$

Where \bar{Y} is the predicted response by the surrogate model constructed with the **normalized** sample, and $\overline{\text{RMSE}}$ is the error calculated with such a surrogate model.

4.1.4 The Random Cross validation Method

When surrogate models are used to facilitate the design process, it is very important to obtain the accuracy information of the surrogate models in terms of model fitting error and model predicting error. While the model fitting error is calculated easily by comparing the true response values and the values predicted by the surrogate model, the model predicting error can be estimated by either one of the following two ways: using an additional random sample, or using re-sampling methods without an additional random sample. Using a random sample is the most reliable way to estimate the model predicting error, but the expense of this approach is very high because one has to run costly physics-based models to obtain the random sample. On the other hand, one would prefer using the costly random sample to construct or improve the surrogate model instead of holding it just for the purpose of error estimation. Therefore, the re-sampling methods are highly preferred. However, the computational expense of the conventional re-sampling methods such as cross validation and bootstrap is still substantial. A new re-sampling method, called random cross validation (RCV), is formed in this research. The scheme of this random cross validation method is show in Figure 4-2. Note again that in this research the surrogate models are limited to RSM, SVR and the hybrid RSSVR.

In this scheme, the surrogate model MM_{Final} is the surrogate model used for design, and the intermediate surrogate model MM_{RCV} is constructed using the same method and pre-specified parameters of MM_{Final} for SVR or the new hybrid method RSSVR.

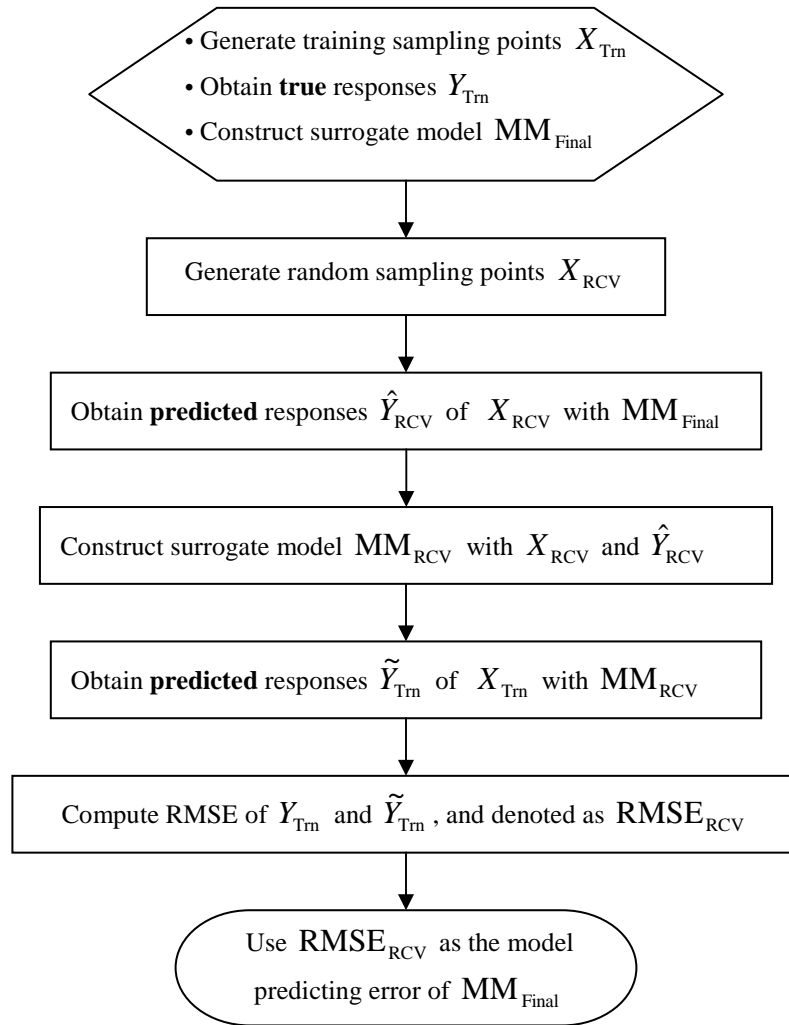


Figure 4-2: Scheme of Random Cross Validation for Estimation of Model Predicting Error

This random cross validation method is computationally cheap comparing with the conventional re-sampling methods because it just needs to execute once. It is found to be able to provide a reasonable estimation of the model predicting errors for the surrogate

models constructed by SVR and the new hybrid surrogate modeling method. Unfortunately, it does not always perform well for surrogate models constructed by RSM.

4.1.5 The Modified AIC and BIC Information Criteria

As discussed previously, a new model selection criterion is needed that makes use of the model fitting error, model predicting error, and model complexity. The original information criteria make use of two kinds of information, i.e. the model fitting error and model complexity measured by the number of parameters in the surrogate model. Since the model predicting error of the surrogate models now can be inexpensively estimated by the random cross validation method, it can be included into the information criteria such that all three kinds of information, i.e. model fitting error, model complexity, and model predicting error, are used. Then those modified information criteria can be used as the desired new model selection criterion.

For RSM or SVR individually, the modified AIC and BIC, denoted as AICC1 and BICC1, are as follows:

$$\text{AICC1} = \frac{1}{2} \left(\ln(\hat{\sigma}_{\text{MLE}}^2) + \ln(\text{RMSE}_{\text{RCV}}^2) \right) + \frac{2m}{s} \quad (4.4)$$

$$\text{BICC1} = \frac{1}{2} \left(\ln(\hat{\sigma}_{\text{MLE}}^2) + \ln(\text{RMSE}_{\text{RCV}}^2) \right) + \frac{m \ln(s)}{s} \quad (4.5)$$

where m is the number of parameters in the model, s is the sample size, and $\hat{\sigma}_{\text{MLE}}^2$ denotes the maximum likelihood estimation of the variance of the residual term. It can be shown that $\hat{\sigma}_{\text{MLE}}^2 = \text{RMSE}_{\text{Tm}}^2$.

$$\hat{\sigma}_{\text{MLE}}^2 = \text{RMSE}_{\text{Tm}}^2 \quad (4.6)$$

For the hybrid surrogate-modeling of RSM and SVR, since the same sample are used twice as shown later, the modified AIC and BIC, denoted as AICC2 and BICC2, are as follows:

$$\text{AICC2} = \frac{1}{2} \left(\ln(\hat{\sigma}_{\text{MLE}}^2) + \ln(\text{RMSE}_{\text{RCV}}^2) \right) + \frac{2(m_{\text{RSM}} + m_{\text{SVR}})}{2s} \quad (4.7)$$

$$\text{BICC2} = \frac{1}{2} \left(\ln(\hat{\sigma}_{\text{MLE}}^2) + \ln(\text{RMSE}_{\text{RCV}}^2) \right) + \frac{(m_{\text{RSM}} + m_{\text{SVR}}) \ln(2s)}{2s} \quad (4.8)$$

where m_{RSM} is the number of parameters in the RSM method, m_{SVR} is the number of parameters in the SVR method.

Just for the purpose of comparison, for the hybrid surrogate-modeling of RSM and SVR, the original AIC and BIC, Equations 2.36 and 2.38, respectively, are adapted as follows:

$$\text{AIC} = \ln(\hat{\sigma}_{\text{MLE}}^2) + \frac{2(m_{\text{RSM}} + m_{\text{SVR}})}{2s} \quad (4.9)$$

$$\text{BIC} = \ln(\hat{\sigma}_{\text{MLE}}^2) + \frac{(m_{\text{RSM}} + m_{\text{SVR}}) \ln(2s)}{2s} \quad (4.10)$$

4.1.6 Model Selection and the Model Selection Advisor

In this research, the task of model selection includes 3 folds: the selection kernel function of SVR, selection of parameters of the surrogate model, and selection of surrogate model structures. These three folds of model selection are executed using different methods.

The kernel function of SVR is pre-selected as the Gaussian radial basis function. The GRBF has received significant attention because of its good performance for various complex problems [35]. This characteristic is very important since usually one does not know the complex relationship between the response and the design variables, and thus it

is very hard to select the best kernel function in advance. For this reason, GRBF is selected as the only kernel function for SVR in this research, although it may not be the best one for a specific problem.

After selecting the kernel function of SVR as the Gaussian radial basis function, it is still very hard to use SVR because there are several general parameters that have to be pre-selected by a user, such as the regularization factor C , the deviation ε if the ε -insensitive loss function is used, and the parameter σ of the GRBF kernel. After these parameters are selected by the user, the method will automatically select the other parameters. For a user who does not know the details of the SVR method, this parameter selection work is difficult. Therefore, selection of these general parameters has to be automated.

The first two general parameters of SVR to be selected are the regularization factor C and the deviation ε . These two can be selected by the practical methods using Equations 2.40 and 2.41, respectively. The third parameter σ has to be selected by the user according to some criterion. Theoretically the structural risk function can be a criterion for selection of the parameter σ . However, because in general it is very hard to calculate the VC dimension, the structural risk function can not be used for this purpose. Instead, either one of the new modified information criteria AICC and BICC is used to select the best parameter σ by minimizing the modified information criterion used, with the aid of an optimizer.

Since all the values of the design variables and responses are normalized, the process to select parameters of SVR is standardized, i.e. for any different problems the computer codes are the same.

The two new modified information criteria are also used to select the best model structure among RSM, SVR, and the hybrid RSSVR. For each response, three candidate surrogate models are constructed using RSM, SVR, and the hybrid method, and the one with minimum value of the information criterion used is chosen as the final surrogate model for design. This model structure selection method is called the model selection advisor.

4.1.7 The Scheme for Hybrid Surrogate-Modeling with RSM and SVR and the Model Selection Advisor

As a solution to the first hypothesis about forming a hybrid method of RSM and SVR, a scheme is provided in Figure 4-3 and Figure 4-4 based on the new techniques and considerations in the previous sub-sections. This scheme includes not only the process to construct a surrogate model using the hybrid method of RSM and SVR, but also model selection from the surrogate models constructed by RSM, SVR, and the hybrid method RSSVR by the model selection advisor discussed above.

For the hybrid method RSSVR, the scheme first uses RSM to fit the model, then uses SVR to fit the errors or residuals between the true responses and the predicted values by the surrogate model just constructed by RSM. The final surrogate model is then the sum of the RSM part and the SVR part.

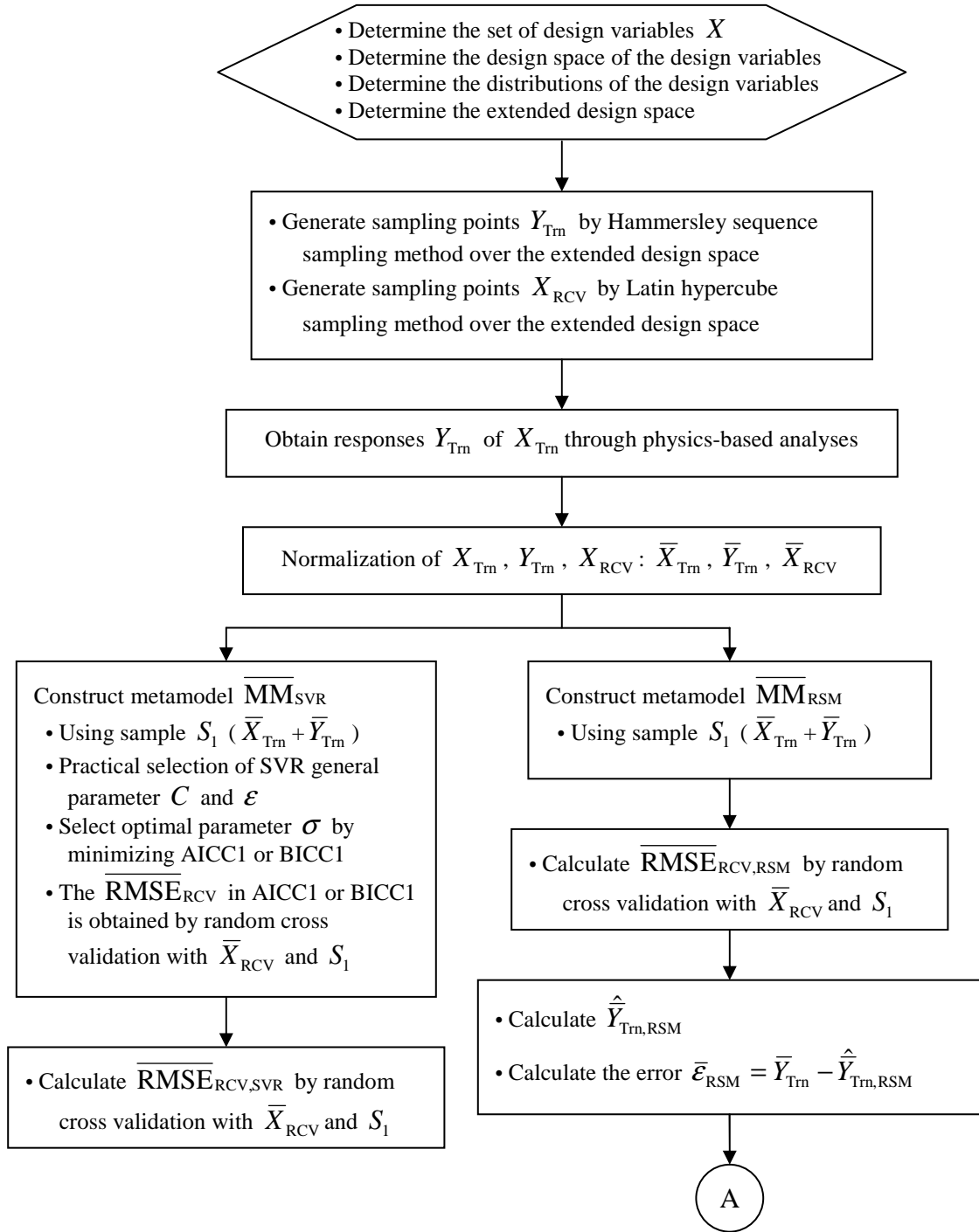


Figure 4-3: Scheme for Hybrid Surrogate-Modeling with RSM and SVR and Model Selection Advisor - I

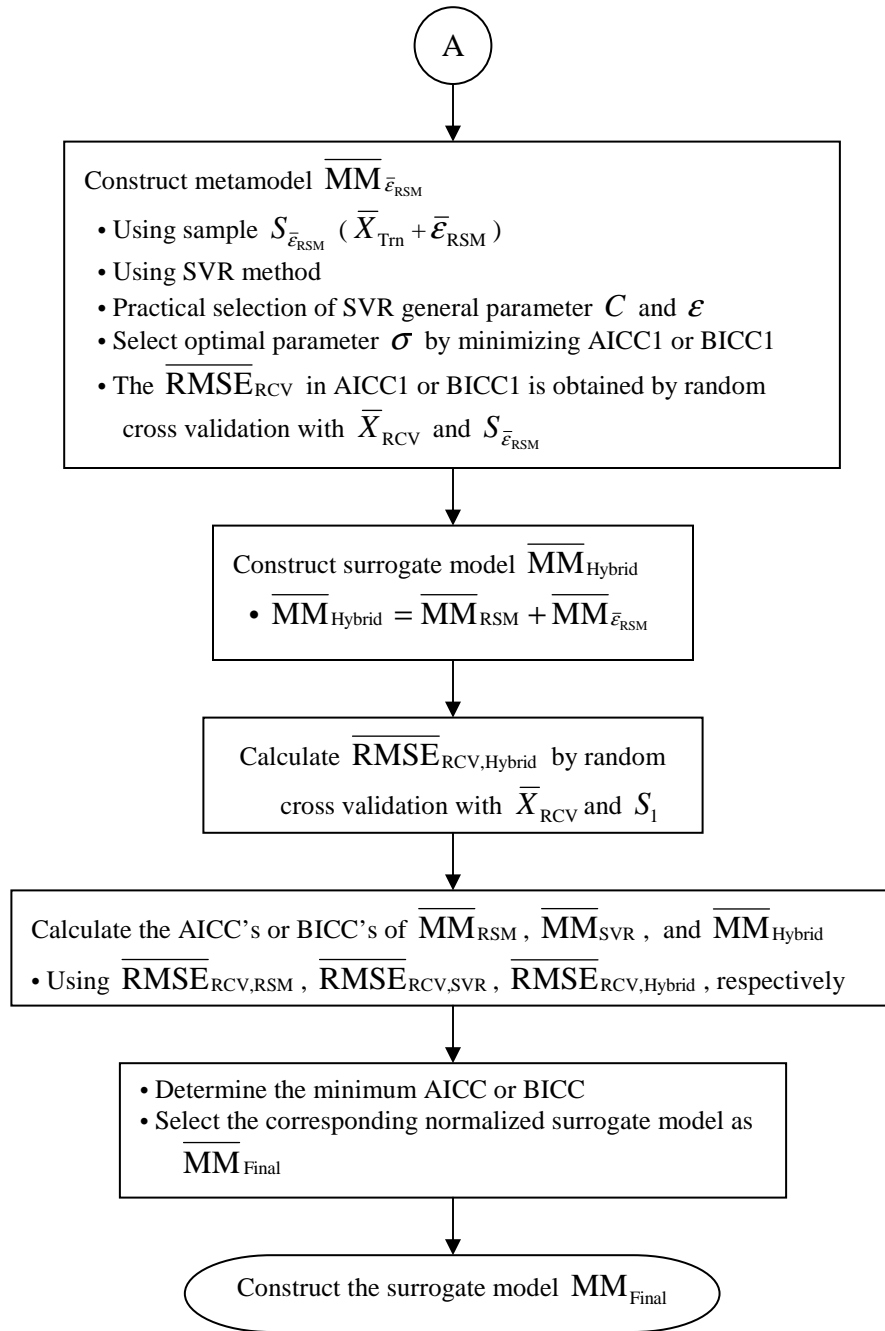


Figure 4-4: Scheme for Hybrid Surrogate-Modeling with RSM and SVR and Model Selection Advisor - II

4.1.8 The Levels of Surrogate Models

The surrogate models for design can be constructed at either disciplinary level or system level. Disciplinary-level surrogate models are recommended if possible. First, for

MDO it is preferred to use surrogate models for disciplinary models than a single global surrogate model for the whole multidisciplinary analysis or simulation. This is because disciplinary surrogate models may lead to less number of analyses or simulations [16]. Second, the relationships among the disciplinary responses and design variables in the resulted disciplinary surrogate models have physical meanings, whereas those relationships may not make sense in the monolithic system-level surrogate models. Third, when there are coupling variables among different disciplines, disciplinary surrogate models are better used or have to be used to assure consistence of coupling variables.

However, sometimes one has to construct system-level surrogate models if there are only monolithic legacy codes available. The monolithic legacy codes, such as FLOPS [100], integrate different disciplines together, and thus can not be used to construct disciplinary surrogate models. In this case, the consistency of coupling variables is assumed to be attained inside the codes.

4.2 Determination of the WPF Solutions under PC's

The realistic conceptual design of complex systems requires solving a joint probabilistic, multi-objective, multidisciplinary optimization problem and finding the WPF solutions for design alternative generation and selection. With the aid of the accurate surrogate models that captures the essence of physics-based models and reduces the computational expense to a manageable level, a new Monte Carlo simulation based method is formed to address this need. This section first introduces the new techniques that are the foundation of this new method; then some considerations about solving the JPMOMDO problem are given; and last the flowchart of the new Monte Carlo simulation based method is provided.

4.2.1 Defining the Neighborhoods for Searching Consistent Designs

As mentioned previously, the multidisciplinary design usually features coupling variables, which result in disjointed consistent design zones in the design space and makes it difficult to find the feasible solutions and WPF. Therefore, the first task of this new Monte Carlo simulation based method is to find an approach to address the problem of disjointed consistent design zones.

As mentioned in Chapter 3 Research Questions and Hypotheses, considering what is needed for a method to solve a JPMOMDO problem and find WPF, this method should:

- 1) Find a MDO solution under deterministic constraints very fast and at low cost, which is consistent for all coupling variables and satisfies all deterministic constraints, denoted as valid solution;
- 2) Explore the entire design space without missing any disjointed consistent design zones;
- 3) On top of requirement 2), be able to find the WPF under deterministic constraints over each disjointed consistent design zone, i.e. local deterministic WPF, if any;
- 4) Find enough and evenly distributed points for each local deterministic WPF;
- 5) On top of all above, find the global WPF under probabilistic constraints, i.e. global probabilistic WPF over the whole design space;
- 6) Find enough and evenly distributed points for the global probabilistic WPF.

The current methods can combine together some of MDO, MOO, and JPA, but not all of the three. Usually those methods adopt a nesting-loop approach with each loop to handle either MDO, or MOO, or JPA. Although this approach has been successfully

applied to many problems and the computational time and cost are acceptable since there are at most two loops nested together, it will result in unacceptable long computational time and high cost if solving a JPMOMDO problem and finding the WPF since there will be three loops nested together and both computational time and cost increase exponentially with the number of loops.

One may consider using domain spanning search methods such as grid search and random search methods since those methods do not have the problem of nesting loops. The grid search method establishes a grid network in the design space and uses the grid knots as search points. This method can guarantee a uniform distribution of the search points over the design space, but the difficulty comes from choosing the appropriate fineness of the grid. With a coarse grid network, one may not obtain any consistent designs in some consistent design zones; another problem is that one may not obtain enough consistent design points to find the WPF with a certain confidence. These two problems are illustrated with a two design variable example in Figure 4-5. On the other hand, with a fine grid network, the computational time may be unacceptable, especially for high dimensional design problems.

The random search method generates random search (design) points and checks the convergence criteria for the coupling variables. There are two main problems with this random search method. First, because the search points are randomly generated, it is possible that only a very few of those points, if not none, are found to be consistent designs. Second, one does not know when to stop the search, because it is hard, if at all, to make randomly generated search points uniformly distribute over the whole design

space since one should not control the random number generator. For these reasons, the domain spanning search methods are not adopted in this research.

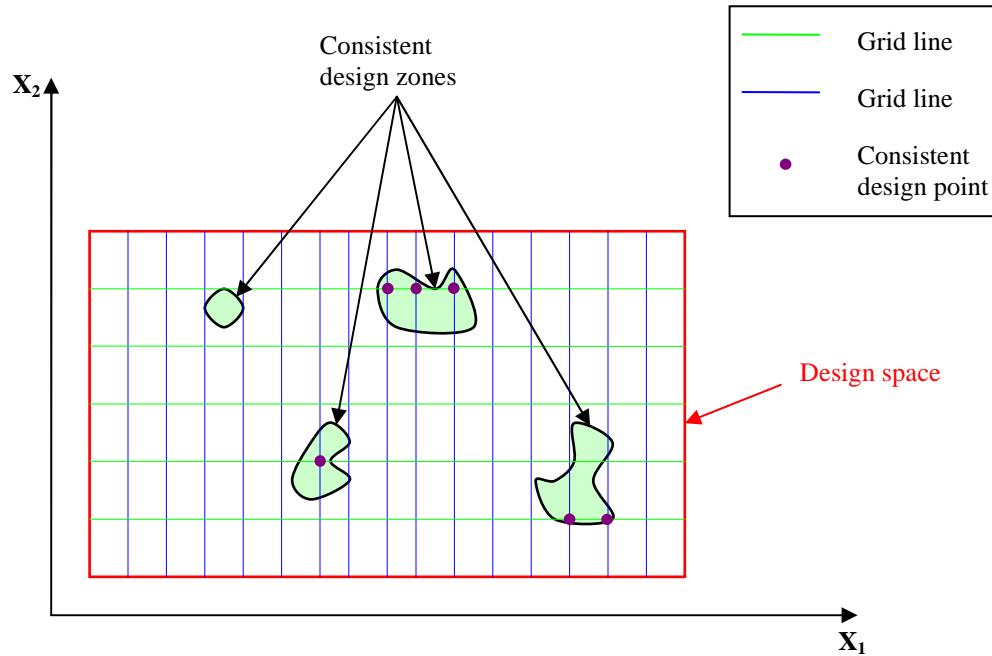


Figure 4-5: Example to Show the Problems with Grid Search

However, combining the ideas of grid search and random search together and making modifications, a new neighborhood search method is formed to efficiently search for consistent design points. The basic idea of this method is to search the neighborhoods of search starting points using an optimizer. This idea is inspired by the fact that most optimizers require a starting point and low and upper bounds of the design variables. One can imagine that the low and upper bounds define a hypercube in a n dimensional space. This approach starts with a set of initial search points that are generated over the design space by uniform Monte Carlo sampling, thus those points are randomly distributed while cover the whole design space uniformly at the same time. This set of sampling points is denoted as sampling points S_2 . Then a hypercube is defined to which each starting point

is centered. The hypercube of each starting point is called the neighborhood of this point. The sizes of the neighborhoods should be defined such that the neighborhoods can cover the entire design space, although there may be overlapping among those neighborhoods. Then an optimizer is used to search for a consistent design point within this hypercube. As long as the neighborhood of a starting point overlaps with a consistent design zone, the optimizer usually can find a consistent design either on the boundary of this consistent design zone or inside it, depending on the performance of the optimizer used. Figure 4-6 illustrates the idea of this approach. This approach is called Monte Carlo simulation based neighborhood search method.

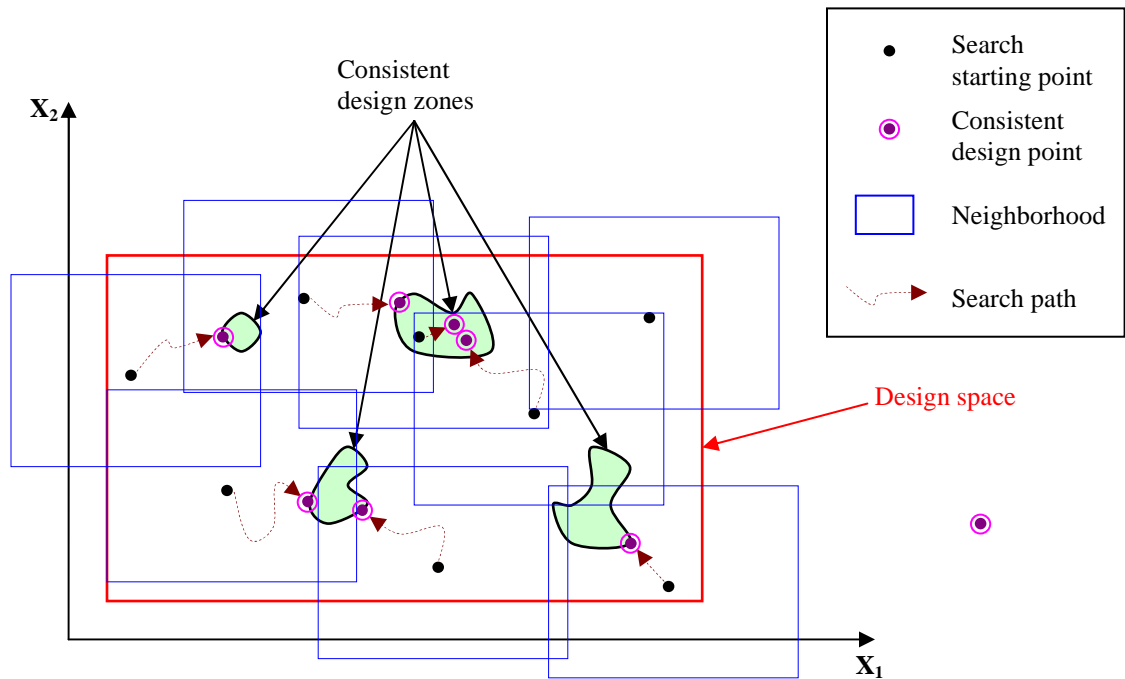


Figure 4-6: Illustration of the Neighborhood Search Method

Whether the neighborhoods of the sampling points S_2 can cover the entire design space depends on two factors: the volumes of the hypercubes and the distribution of the

sampling points. If the sampling points are uniformly distributed within the design space, then each hypercube can occupy $\frac{1}{s_2}$ times the total volume of the design space, where s_2 is the size of S_2 ; and the length of i^{th} side of the hypercube can be $\frac{1}{\sqrt[n]{s_2}} \Delta x_i$, where Δx_i is the range of i^{th} design variable. However, in order to allow some non-uniform distribution of the starting points, a larger length of i^{th} side of the hypercube for overlapping is given as

$$a_i = \frac{2}{\sqrt[n]{s_2}} \Delta x_i, i = 1, \dots, n \quad (4.11)$$

where a_i is the length of i^{th} side of the hypercube, Δx_i is the range of i^{th} design variable.

Denotes a starting point as X_0 , then the lower and upper bounds of i^{th} design variable in a neighborhood search problem is given as

$$x_{i0} - \frac{a_i}{2} \leq x_i \leq x_{i0} + \frac{a_i}{2}, i = 1, \dots, n \quad (4.12)$$

One note is that if a part of a neighborhood is not within the design space, this part should be cut off.

This neighborhood search method has many advantages. First, unlike the modified Normal Constraint method, no optimization effort is used to generate initial search points.

Second, like the modified NC, this method also guarantees evenly distributed global PF points over the whole design space. This method combines the flexibility of the random search method and uniformity of the grid search method. Since this set of initial search points is generated over the design space by uniform Monte Carlo sampling, those

points are randomly distributed while covering the entire design space uniformly at the same time. Since the initial search points are uniformly distributed, under this mechanism no disjointed consistent design zone will be missed as long as the neighborhoods will cover the entire design space. Usually an optimizer will reach a different consistent design if starting from a different initial search point. Since the initial search points are randomly distributed, under this mechanism the consistent design solutions will be randomly distributed and some of those solutions will be on or near the local deterministic WPF's. Since each neighborhood is small, it is almost impossible for a neighborhood to contain more than one local deterministic optimal design. Therefore, if the optimizer is used to directly search for a local deterministic WPF point of the part of a consistent design zone within a neighborhood, it will almost surely find such a point. Therefore, the global (probabilistic) WPF can be found by either indirectly or directly searching for local deterministic WPF's, and the representing discrete points of the global WPF will be evenly distributed. One note is that the coupling variables are included into the design variables of the optimizer, as can be seen later, and by doing so the multidisciplinary problem is decoupled. From the above description one can see that this neighborhood search method solves a MDO problem in a way almost the same as the OBD method in the sense of decoupling based on an optimizer.

The third advantage of is that the required number of initial search points can be estimated since it is based on Monte Carlo simulation. If one thinks the instance of obtaining a global WPF is probabilistic, then the required number of initial search points can be estimated based on Equation 2.44. The rule-of-thumb equation will be given later.

In summary, this new Monte Carlo simulation Based neighborhood search method can satisfy the requirements to solve a JPMOMDO problem and find the WPF because of the characteristics and advantages described above.

Undeniably, this neighborhood search method requires a large number of initial search points since it is based on Monte Carlo simulation. However, this large number of initial search points is necessary for finding a global WPF under probabilistic constraints. On the other hand, one may suspect this method requires unacceptable computational time since there are such a large number of neighborhoods to be searched by an optimizer. However, since each neighborhood is small and usually contains one local optimal design, the optimization process is fast in a neighborhood. Therefore, the overall computational time is not long but manageable.

4.2.2 Two Schemes for the Neighborhood Search Method

Although the neighborhoods are defined and used to search for consistent design points, it needs some special schemes to wisely use this concept to find the consistent designs because the number of consistent designs is infinite and not all consistent design solutions are feasible, satisfying both the **deterministic and probabilistic** constraints. The goals of such a scheme are to find valid designs (points), and the valid designs it has found include some designs being the WPF points or designs near to the WPF points. Some valid designs are also the (probabilistic) WPF points; some ones are close to the WPF points and are denoted as near WPF design solutions. Two Monte Carlo simulation based schemes are formed for the above goals.

First of all, there is one fundamental requirement for a satisfactory scheme: the scheme must be able to find the optimal valid design solutions for single-objective

optimization problems with deterministic constraints, or design solutions very near to these optimal single-objective ones. This is because the design solutions for these single-objective optimization problems can be fair easily found by an existing single-objective optimizer. If one scheme can not find such design solutions or near ones, obviously it is not suitable.

Since the two schemes are based on Monte Carlo simulation, the instance of obtaining a useful WPF is probabilistic, and thus the size of S_2 can be estimated based on Equation 2.44.

Denote the number of sampling points estimated by Equation 2.44 as s_0 , the number of coupling variables as n_{CX} , and the number of objective functions as n_{OF} , then as a rule of thumb, the size of S_2 is given as

$$s_2 = 2n_C s_0 \quad (4.13)$$

$$n_C = \max(n_{OF}, n_{CX}) \quad (4.14)$$

The first scheme uses the deterministic constraints and sets errors of the coupling variables as objective functions to be minimized, then uses the a multi-objective optimizer, such as the Goal Attainment optimizer “fgoalattain” in Matlab[®], to search one valid design solution for each search starting point, if any. Figure 4-7 shows the first scheme.

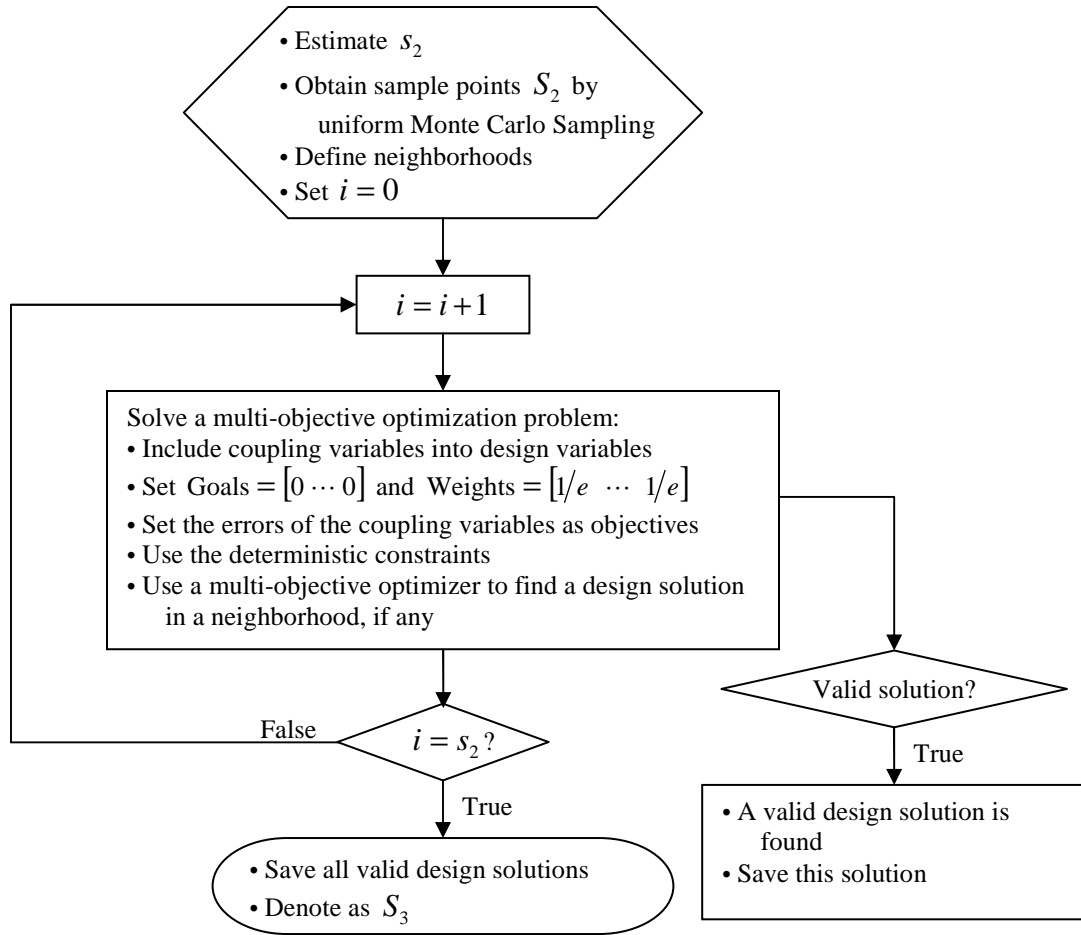


Figure 4-7: First Scheme for the Neighborhood Search Method

The second scheme directly finds the deterministic WPF of the part of a consistent design zone in a neighborhood for each search starting point, if any. This scheme first searches for the optimal valid design solutions of e single-objective optimization problems, if any. The objective function of each single-objective optimization problem is one of the e original objectives. The constraints of each of the single-objective optimization problem include all the deterministic constraints and n_{CX} convergence conditions for the coupling variables. Then the above e objective function values are set as the goals for a multi-objective optimization problem. The objective functions of this multi-objective optimization problem are the e original objectives, and the constraints

are all the deterministic constraints and the convergence conditions for the coupling variables. Note that the optimal valid design solutions of the single-objective optimization problems are also deterministic WPF solutions. By this way, the deterministic WPF of the part of a consistent design zone in a neighborhood is represented by $(e+1)$ points, if any. A single-objective optimizer, such as “fmincon” in Matlab[®], is used for the single-objective optimization problems, and another optimizer such as “fgoalattain” is used for the multi-objective optimization problem. Figure 4-8 shows the basic idea and Figure 4-9 shows the flowchart of the second search scheme.

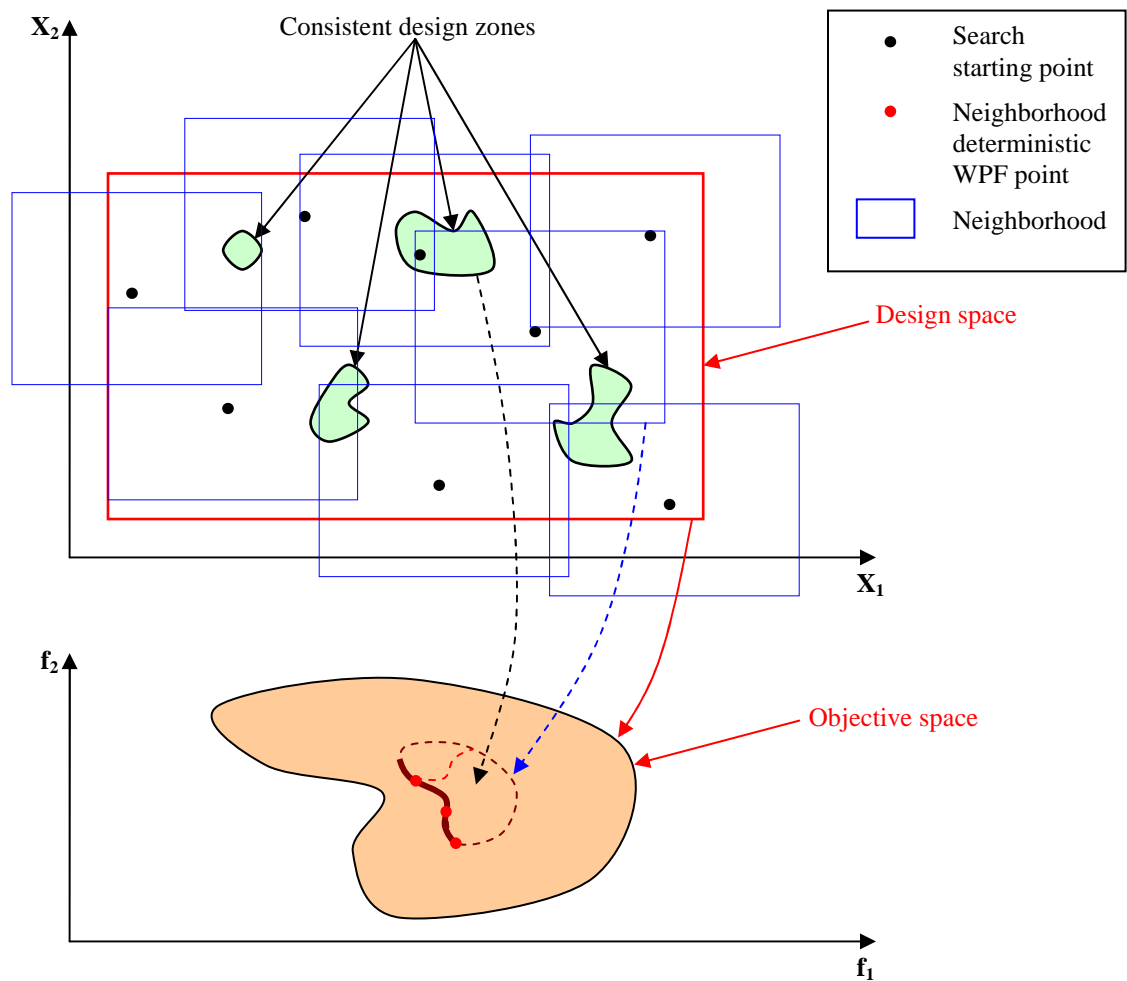


Figure 4-8: Idea of the Second Search Scheme

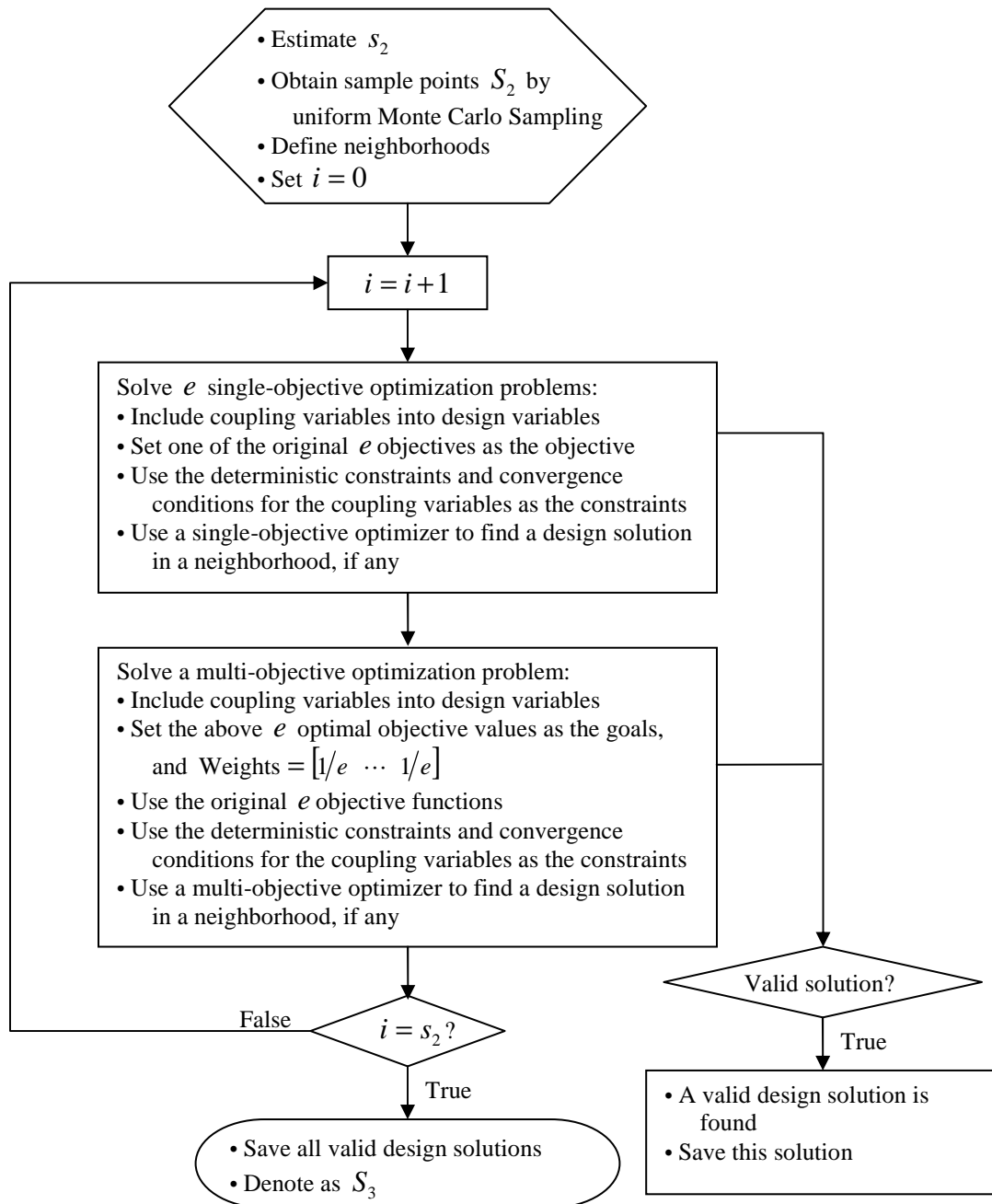


Figure 4-9: Second Scheme for the Neighborhood Search Method

There are two reasons to form these two schemes. First, if the WPF found by the two schemes are **similar**, one can safely say that the true or nearly true WPF is found, since the two schemes are following different approaches. This is very important because

usually an engineering problem is so complicated that one can not determine the true WPF, or one can not afford to use the original models (the CA's) to find the true WPF. Second, the design solutions found by the two schemes are different. Thus more solutions can be found by using the two schemes at the same time.

4.2.3 Relaxation of Converging Conditions for Coupling Variables and Thresholds in PC's

Because of the errors inherent in the surrogate models, the convergence conditions for the coupling variables and the thresholds in the PC's should be relaxed with some tolerances. These tolerances have to be carefully selected in order to neither exclude too many or even all true valid designs, nor include too many designs that are not true valid designs.

The tolerances can be selected based on the RMSE of the model predicting errors of the surrogate models, whereas the model fitting errors are not appropriate for this purpose. Since the random cross validation method is able to provide a reasonable estimation of the model predicting errors for the surrogate models constructed by the new hybrid surrogate-modeling method, as a rule of thumb, the tolerance is given as

$$\text{tolerance} = 2 \text{RMSE}_{\text{RCV,Hybrid}} \quad (4.15)$$

The pseudo programs for relaxation are shown in Figure 4-10 and Figure 4-11.

```
If error_of_CX ≥ tolerance
    convergence_condition = error_of_CX - tolerance
Else
    convergence_condition = 0
End
```

Figure 4-10: Pseudo Program for Relaxation of Convergence Conditions of Coupling Variables

```

If  $g_j(X) \geq th_j + \text{tolerance}$ 
    constraint_condition =  $g_j(X) - th_j - \text{tolerance}$ 
Else
    constraint_condition = 0
End

If  $\text{absolute}(h_k(X) - th_k) \geq \text{tolerance}$ 
    constraint_condition =  $\text{absolute}(h_k(X) - th_k) - \text{tolerance}$ 
Else
    constraint_condition = 0
End

```

Figure 4-11: Pseudo Programs for Relaxation of Constraint Conditions in PC's

4.2.4 The Flowchart for Determination of the WPF Solutions of a JPMOMDO Problem

One basic assumption for this new framework of determination of the WPF solutions of a JPMOMDO in this research is: if the search starting points S_2 are randomly uniformly generated (uniform Monte Carlo sampling) and the size of S_2 is large enough, e.g. the sample size estimation given by Equation 4.13, the valid designs found by the neighborhood search method can include some WPF design solutions or near WPF design solutions. This is because when the number of sample points of S_2 is large, the neighborhoods are small, and the values of the responses of the consistent designs in a neighborhood, if any, are approximately constant. Under this assumption, if there is a local WPF design solution in a neighborhood, the other valid designs in this neighborhood are very close to this design solution.

This part of flowchart starts with obtaining the valid design solutions S_3 . For each design of S_3 , the values of the design variables are treated as nominal values (such as mean values), and a Monte Carlo sampling is executed according to the distributions of

the design variables about this point of S_3 . The sample size estimation is given by Equation 2.44. The resulted sample points are denoted as S_4 .

Each design point of S_4 is checked to see if it is a valid design, and the probabilities of satisfying each PC (probability of success, POS) are calculated for all valid design points of S_4 using the counting Equation 2.56. Then probabilities of the valid design points of S_4 are used to check if those points satisfy the PC's jointly. If the all the PC's are satisfied jointly, the corresponding design solution of S_3 is saved as a candidate design, which is a feasible design. A candidate design solution is a valid design solution of S_3 of which random sample points S_4 result in satisfaction of all PC's jointly. The resulting candidate points are denoted as S_5 .

Then the multiple objective values of the candidate points S_5 are evaluated, and are used to discover the points on the WPF in the objective space.

As the last step, the design solutions of S_5 corresponding to these WPF points are located accordingly, and are denoted collectively as S_{WPF} .

The flowchart is shown in Figure 4-12.

A final note is that the thresholds '0' in the definition of the WPF point can be relaxed with a positive value such that more near WPF points can be obtained.

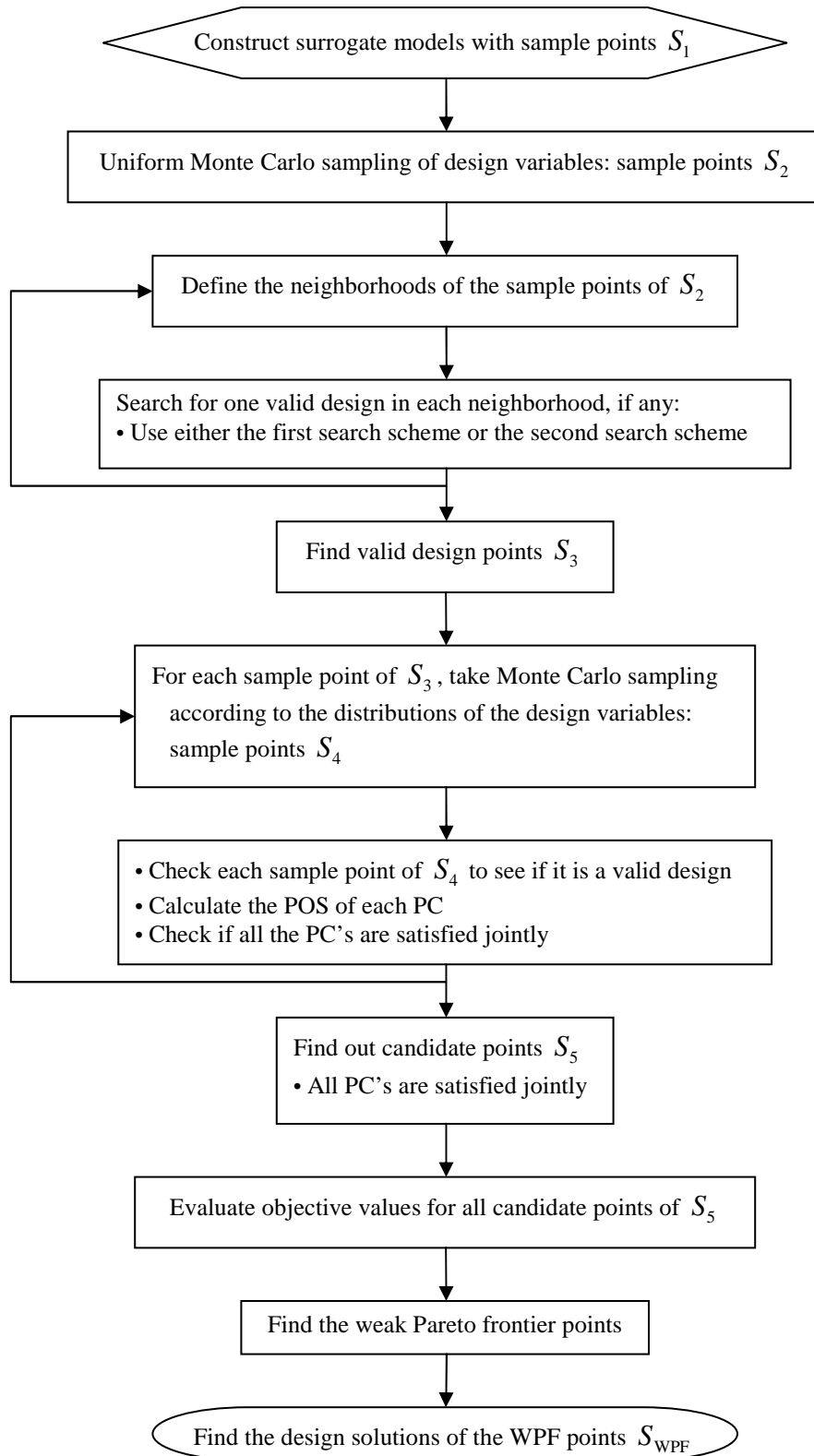


Figure 4-12: Flowchart for Determination of the WPF Solutions under PC's

5 IMPLEMENTATION AND RESULTS

Three pure mathematical examples are used to demonstrate the capacity of SVR and the new hybrid surrogate-modeling method to fit given models. The qualities of the surrogate models of the three methods RSM, SVR, and hybrid method are also compared quantitatively and visually. The hybrid method is also compared with the Neural Network method for those three examples. Three two-objective and one three-objective deterministic optimization problems are used to demonstrate that this framework can surely find the true weak Pareto frontier, although only the second search scheme is used since the first search scheme can not be used without coupling variables. A simple yet typical aircraft design problem and a simple yet typical reusable launch vehicle design problem are solved to demonstrate the feasibility of this new framework for determination of the WPF solutions under PC's. Here 'simple' just means the disciplinary analyses are formulated with explicit equations, which do not exist in a real design process.

5.1 Pure Mathematical Examples of Surrogate Modeling

The three pure mathematical examples given here are the hemisphere, wave function, and Rastrigin function. Those examples are selected in this research because those examples have different orders of nonlinearity, numbers of local extremes, and global behaviors. All these three examples have two design variables and can be visualized. For each example, the three different surrogate-modeling methods RSM, SVR, and the hybrid method are used to construct the surrogate models, and the results are

compared. The Neural Network method is also compared with the hybrid method with the three examples above.

5.1.1 The Upper Hemisphere Example

For this example of upper hemisphere, three cases are considered:

Case 1, the values of the design variables and the responses are not normalized, the kernel is exponential radial basis function (ERBF), and the general parameters of SVR C , ε and σ are given by trial and error;

Case 2, the values of the design variables and responses are not normalized, the kernel is GRBF, the parameters C and ε are estimated by the practical estimation method, and the parameter σ is given by trial and error;

Case 3, the values of the design variables are normalized while the response values are not, the kernel is GRBF, the parameters C and ε are estimated by the practical estimation method, and the parameter σ is selected by minimizing the modified information criterion BICC.

For each case, all the three surrogate-modeling methods are used to construct surrogate models. For each case, the training sample for surrogate model construction includes 100 points by HS sampling, the sample for estimation of the true model predicting error (MPE) includes 200 points by LHC sampling, and the sample for RCV includes 200 points by LHC sampling. The upper hemisphere is given as:

$$y = \sqrt{100 - x_1^2 - x_2^2}, \quad x_1, x_2 \in [-10, 10]$$

Figure 5-1 shows the upper hemisphere over the given design space.

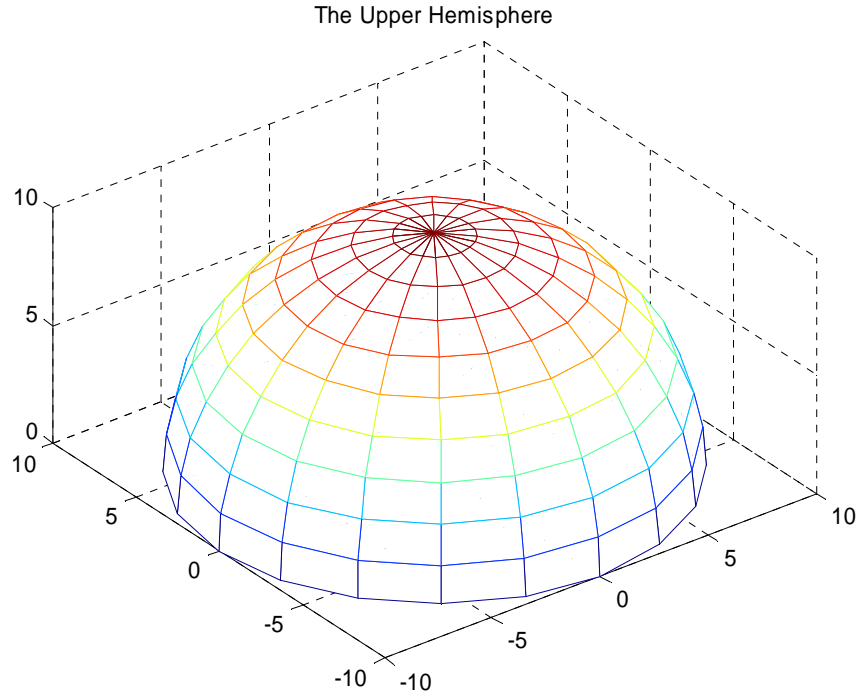


Figure 5-1: Illustration of the Upper Hemisphere

The values of the general parameters and goodness of fit are listed in Table 4, and the visualization of the resulted surrogate models is shown in Figure 5-2 – Figure 5-8.

Table 4: Values of General Parameters and Goodness of Fit for the Upper Hemisphere

Case	Method	Parameters of SVR	R^2	RMSE _{Tm}	RMSE _{RCV}	RMSE _{MPE}
1	RSM		0.96083	0.49266	0.33693	0.45627
	SVR	$C = 200; \epsilon = 0.01; \sigma = 2$	1.00000	0.00995	0.34065	0.47713
	Hybrid	$C = 200; \epsilon = 0.01; \sigma = 2$	0.99999	0.00981	0.22215	0.31217
2	RSM		0.96083	0.49266	0.34768	0.50033
	SVR	$C = 14.05; \epsilon = 0.0158; \sigma = 1.5$	0.99998	0.01584	0.29786	0.39346
	Hybrid	$C = 1.487; \epsilon = 0.0198; \sigma = 1.5$	0.99723	0.13895	0.20400	0.29038
3	RSM		0.96083	0.49266	0.20382	0.49182
	SVR	$C = 14.05; \epsilon = 0.0158; \sigma = 0.081$	0.99998	0.01568	0.11946	0.41626
	Hybrid	$C = 1.487; \epsilon = 0.0198; \sigma = 0.027$	0.99990	0.02525	0.35913	0.42541

RSM - Upper Hemisphere

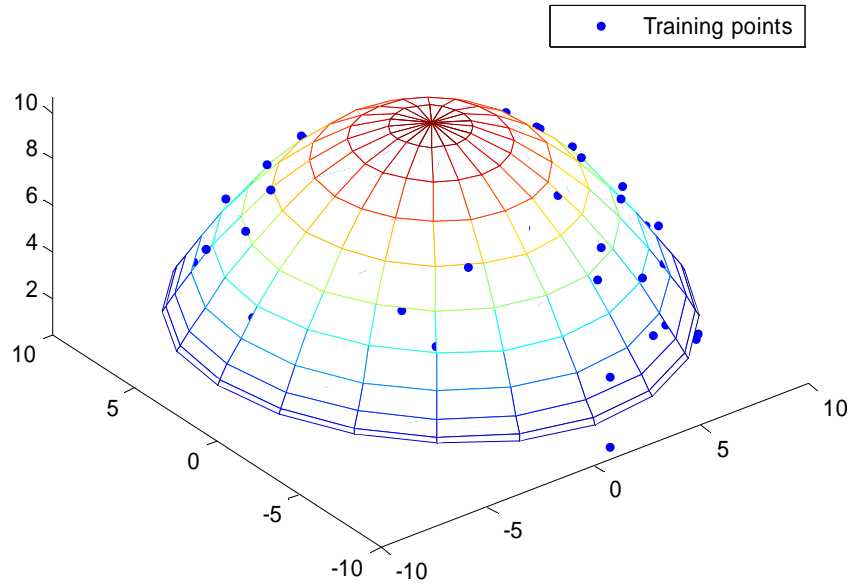


Figure 5-2: Surrogate Model for the Upper Hemisphere by RSM – Case 1, 2, 3

SVR - Upper Hemisphere

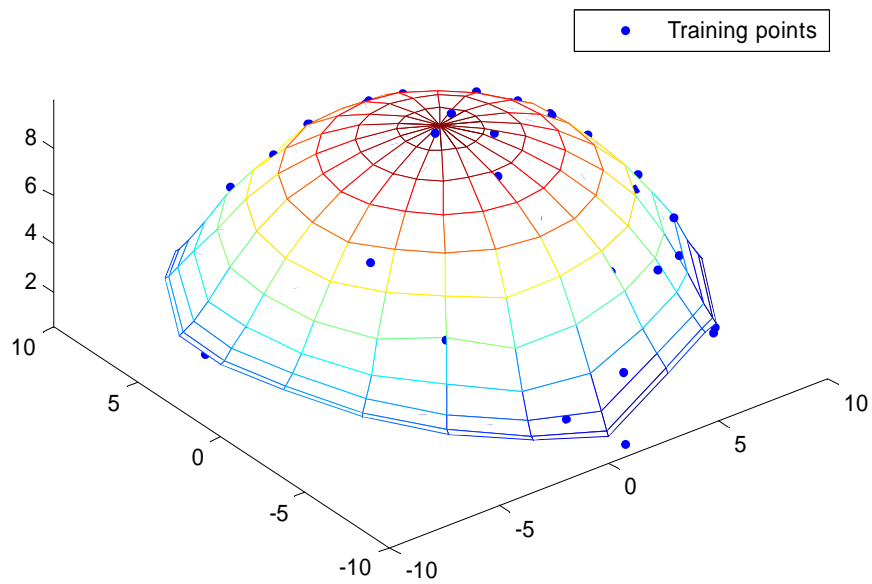


Figure 5-3: Surrogate Model for the Upper Hemisphere by SVR – Case 1

Hybrid - Upper Hemisphere

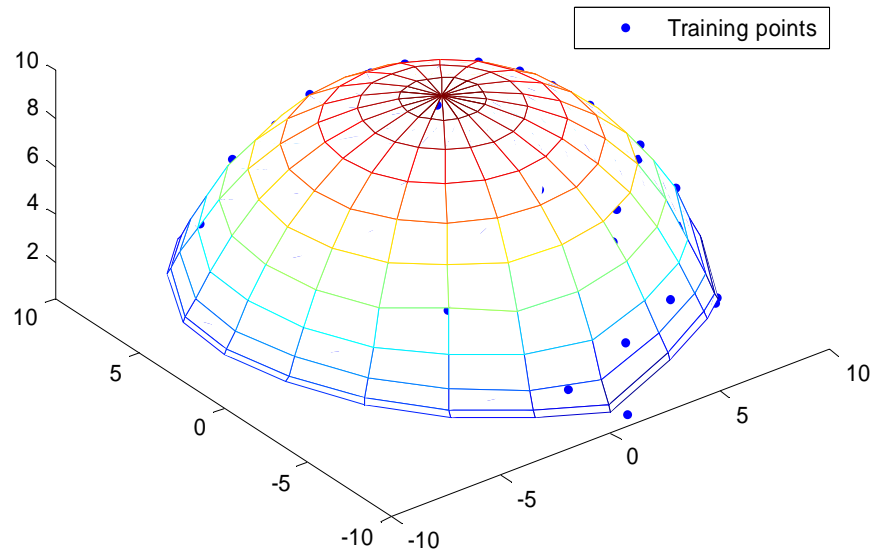


Figure 5-4: Surrogate Model for the Upper Hemisphere by Hybrid – Case 1

SVR - Upper Hemisphere

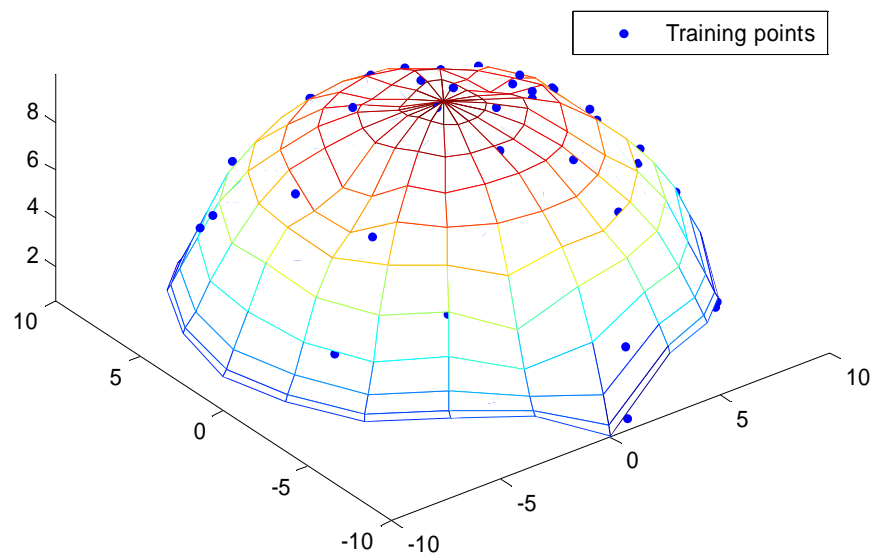


Figure 5-5: Surrogate Model for the Upper Hemisphere by SVR – Case 2

Hybrid - Upper Hemisphere

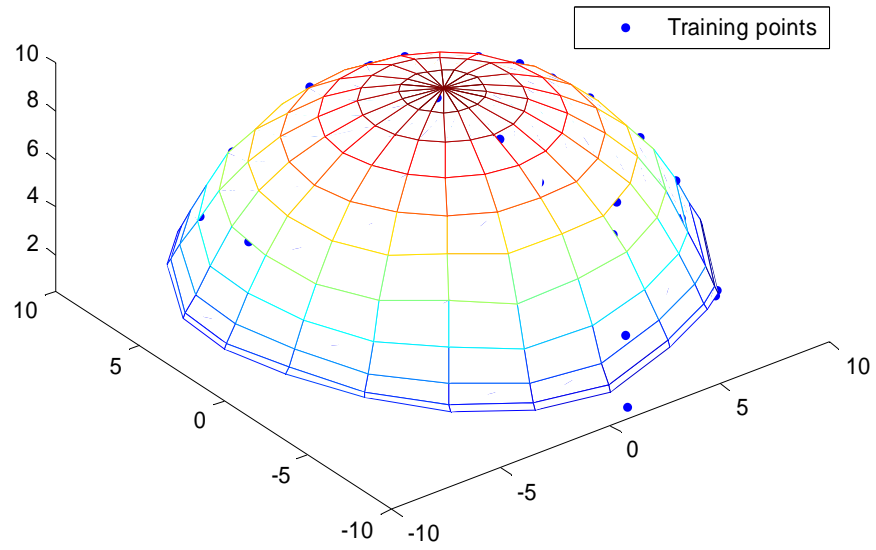


Figure 5-6: Surrogate Model for the Upper Hemisphere by Hybrid – Case 2

SVR - Upper Hemisphere

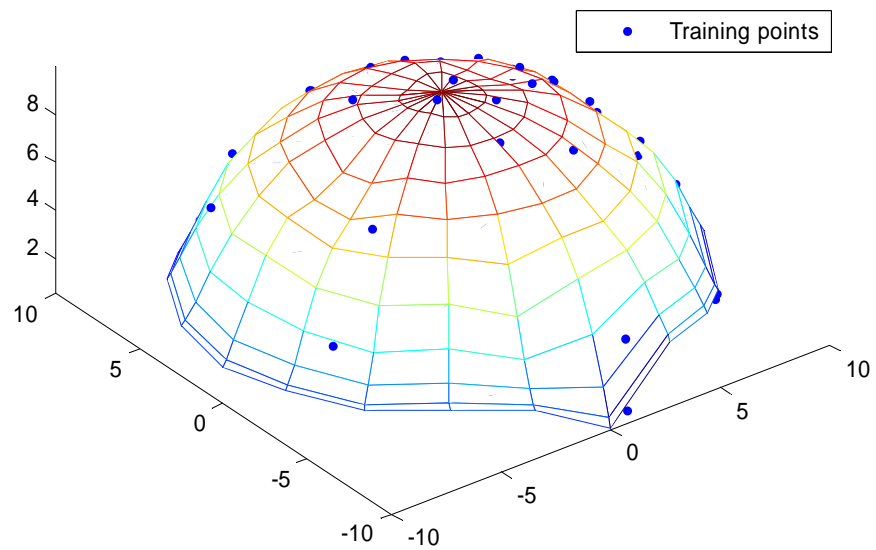


Figure 5-7: Surrogate Model for the Upper Hemisphere by SVR – Case 3

Hybrid - Upper Hemisphere

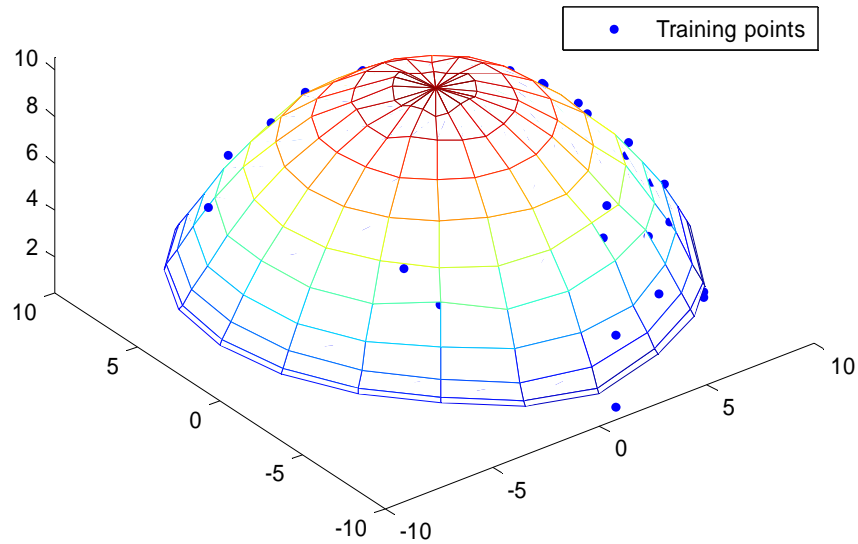


Figure 5-8: Surrogate Model for the Upper Hemisphere by Hybrid – Case 3

5.1.2 The Wave Function Example

For this example of wave function, three cases are considered:

Case 1, the values of the design variables and the responses are not normalized, the kernel is ERBF, and the general parameters of SVR C , ε and σ are given by trial and error;

Case 2, the values of the design variables and responses are not normalized, the kernel is GRBF, the parameters C and ε are estimated by the practical estimation method, and the parameter σ is given by trial and error;

Case 3, the values of the design variables are normalized while the response values are not, the kernel is GRBF, the parameters C and ε are estimated by the practical estimation method, and the parameter σ is selected by minimizing the modified information criterion BICC.

For each case, all the three surrogate-modeling methods are used to construct surrogate models. For each case, the training sample for surrogate model construction includes 120 points by HS sampling, the sample for estimation of true MPE includes 200 points by LHC sampling, and the sample for RCV includes 200 points by LHC sampling. The wave function is given as:

$$y = (x_1^2 - x_2^2) \sin(x_1/2), \quad x_1, x_2 \in [-10, 10]$$

Figure 5-9 shows the wave function over the given design space.

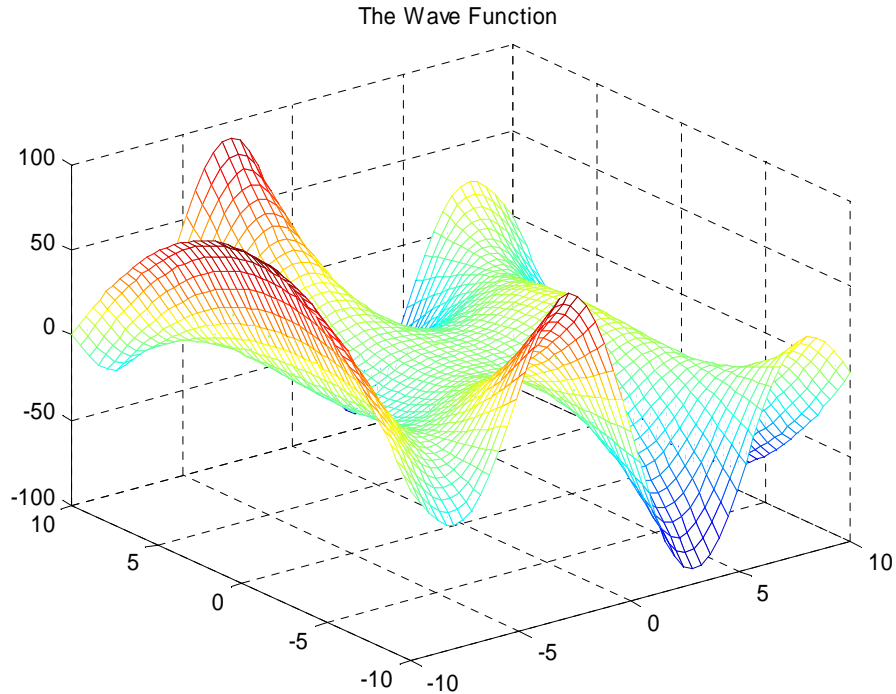


Figure 5-9: Illustration of the Wave Function

The values of the general parameters and goodness of fit are listed in Table 5, and the visualization of the resulted surrogate models is shown in Figure 5-10 – Figure 5-16.

Table 5: Values of General Parameters and Goodness of Fit for the Wave Function

Case	Method	Parameters of SVR	R^2	RMSE _{Tm}	RMSE _{RCV}	RMSE _{MPE}
1	RSM		0.39492	25.27269	21.12868	26.73301
	SVR	$C = 200; \epsilon = 0.01; \sigma = 1.5$	1.00000	0.01000	6.89083	6.38921
	Hybrid	$C = 200; \epsilon = 0.01; \sigma = 1.5$	1.00000	0.01000	6.83087	6.03544
2	RSM		0.39492	25.27269	25.39322	28.41188
	SVR	$C = 98.01; \epsilon = 0.0088; \sigma = 1.5$	0.99997	0.17855	3.82165	4.79405
	Hybrid	$C = 76.14; \epsilon = 0.0116; \sigma = 1.5$	0.99998	0.14097	3.40355	4.21043
3	RSM		0.39492	25.27269	25.02884	26.95390
	SVR	$C = 98.01; \epsilon = 0.0088; \sigma = 0.069$	1.00000	0.00875	2.84765	6.38358
	Hybrid	$C = 76.14; \epsilon = 0.0116; \sigma = 0.069$	1.00000	0.01155	4.81235	4.73529

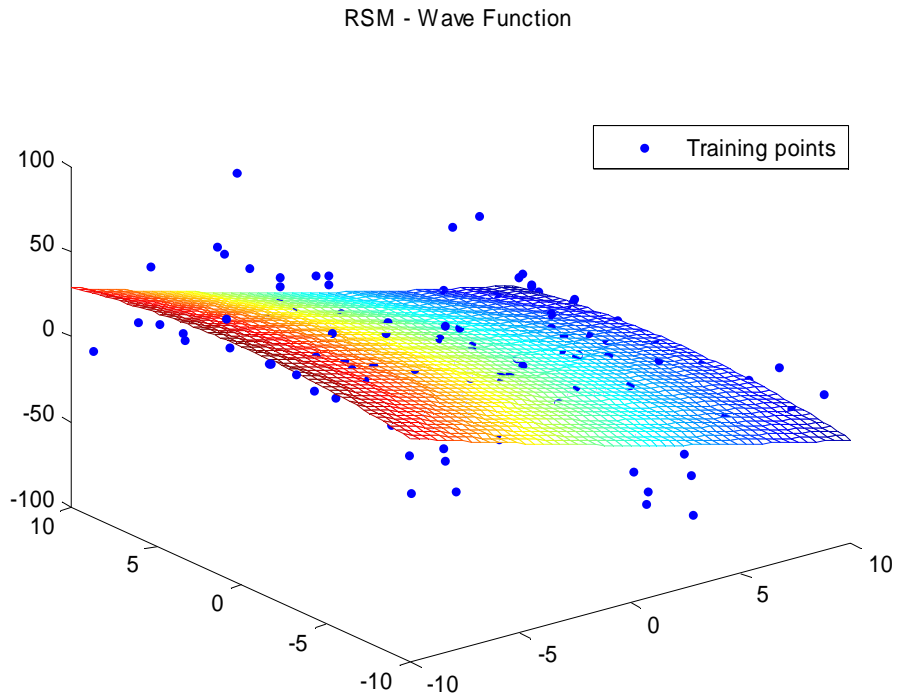


Figure 5-10: Surrogate Model for the Wave Function by RSM – Case 1, 2, 3

SVR - Wave Function

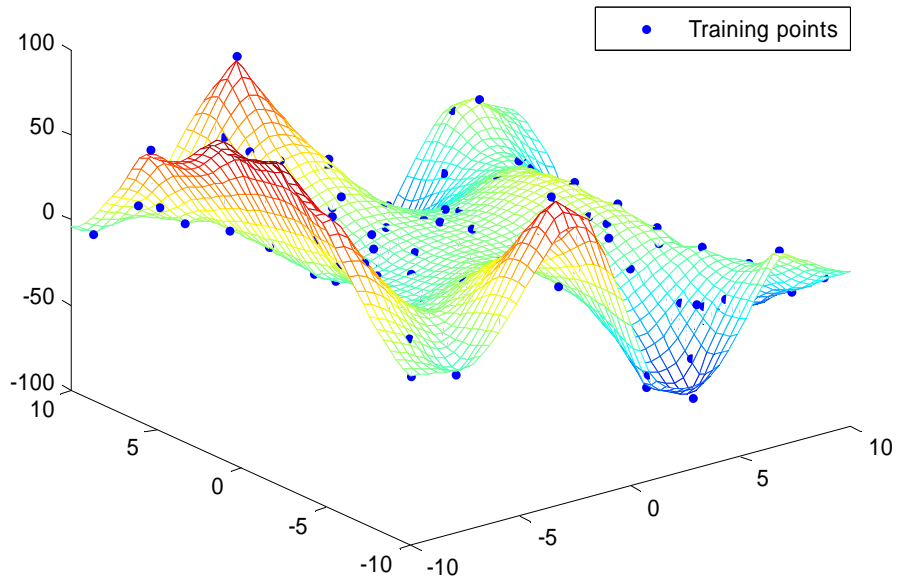


Figure 5-11: Surrogate Model for the Wave Function by SVR – Case 1

Hybrid - Wave Function

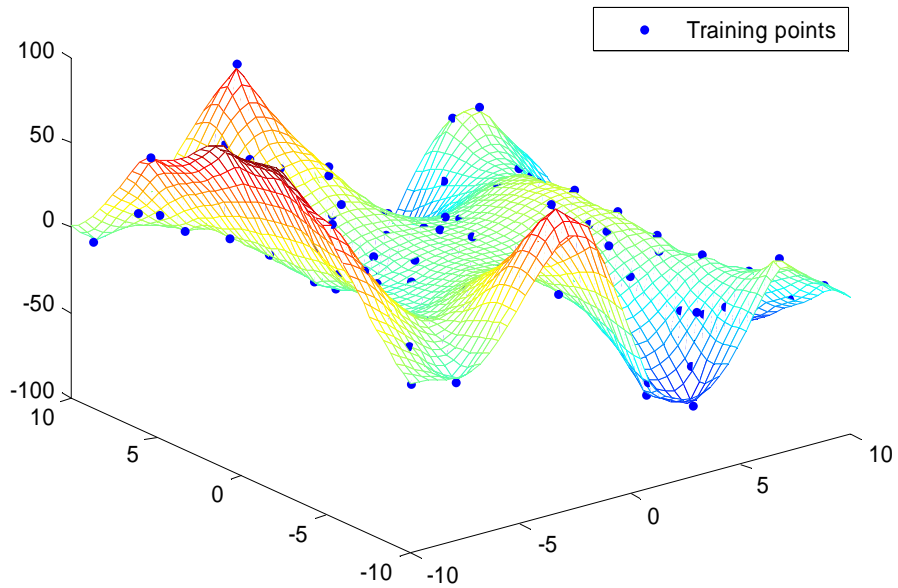


Figure 5-12: Surrogate Model for the Wave Function by Hybrid – Case 1

SVR - Wave Function

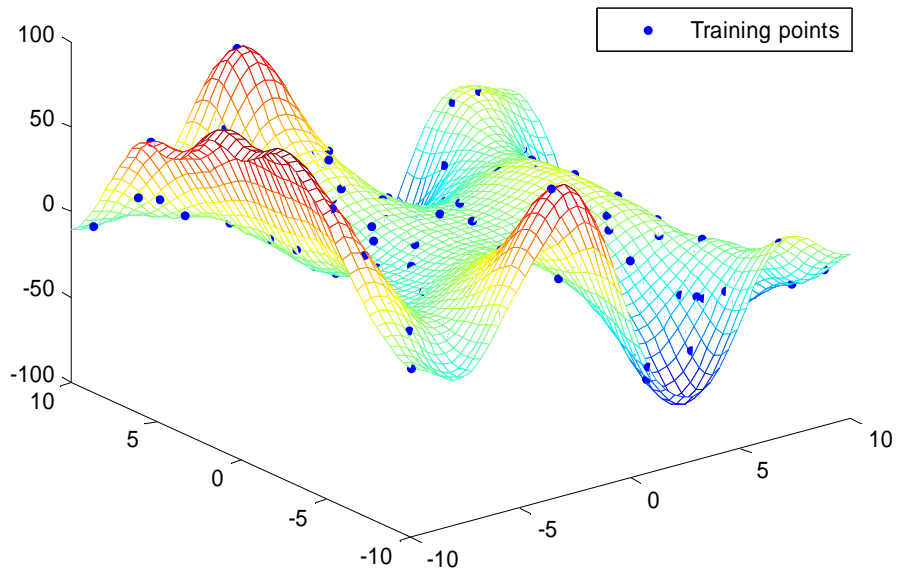


Figure 5-13: Surrogate Model for the Wave Function by SVR – Case 2

Hybrid - Wave Function

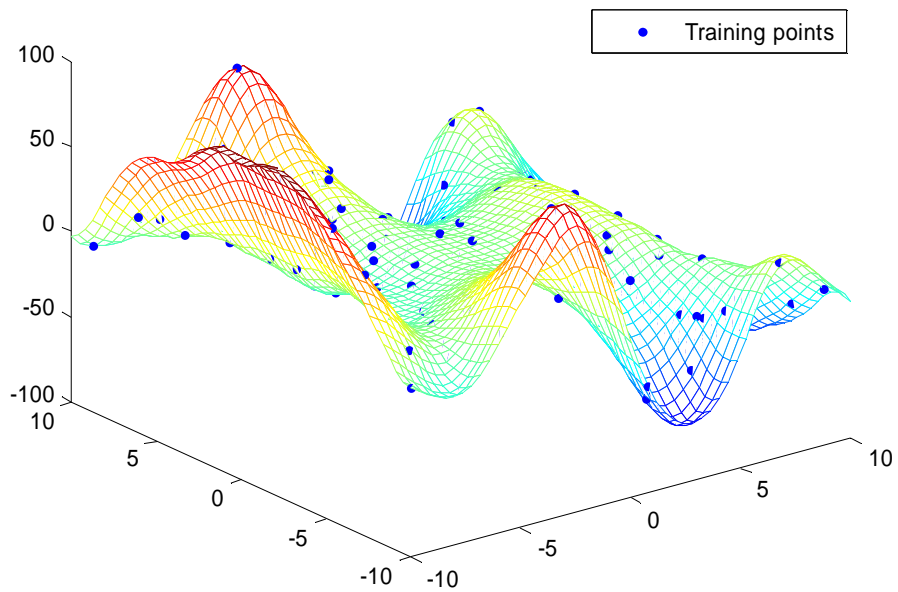


Figure 5-14: Surrogate Model for the Wave Function by Hybrid – Case 2

SVR - Wave Function

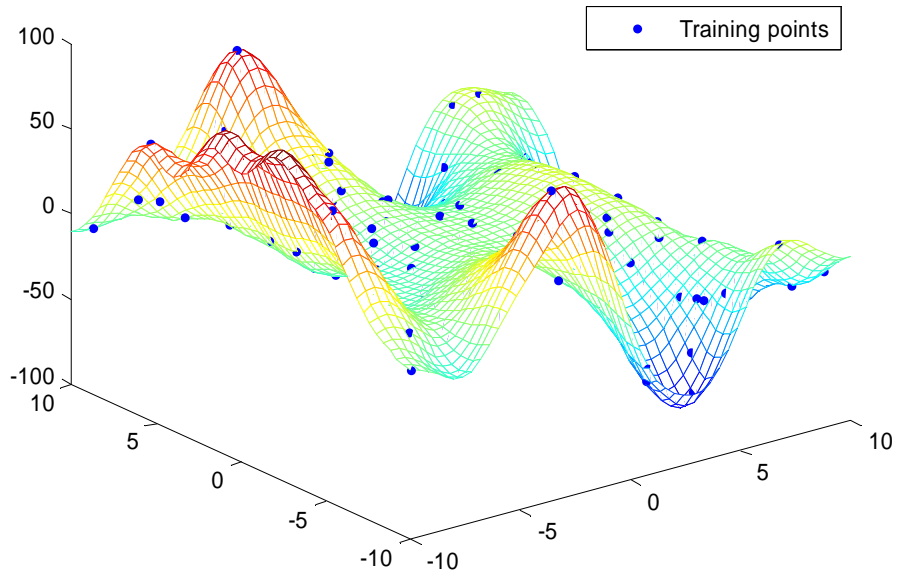


Figure 5-15: Surrogate Model for the Wave Function by SVR – Case 3

Hybrid - Wave Function

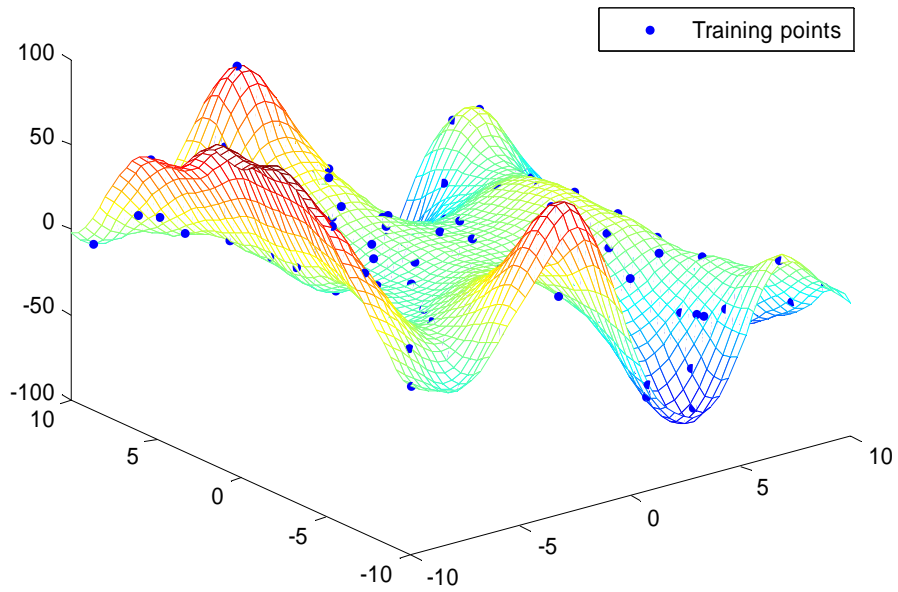


Figure 5-16: Surrogate Model for the Wave Function by Hybrid – Case 3

5.1.3 The Rastrigin Function with 36 Peaks Example

For this example of the Rastrigin function with 36 peaks, three cases are considered:

Case 1, the values of the design variables and the responses are not normalized, the kernel is ERBF, and the general parameters of SVR C , ε and σ are given by trial and error;

Case 2, the values of the design variables and responses are not normalized, the kernel is GRBF, the parameters C and ε are estimated by the practical estimation method, and the parameter σ is given by trial and error;

Case 3, the values of the design variables are normalized while the response values are not, the kernel is GRBF, the parameters C and ε are estimated by the practical estimation method, and the parameter σ is selected by minimizing the modified information criterion BICC.

For each case, all the three surrogate-modeling methods are used to construct surrogate models. For each case, the training sample for surrogate model construction includes 120 points by HS sampling, the sample for estimation of true MPE includes 200 points by LHC sampling, and the sample for RCV includes 200 points by LHC sampling. The Rastrigin function with 36 peaks and consequently 25 troughs is given as:

$$y = A \cdot n + \sum_{i=1}^n x_i^2 - \sum_{i=1}^n A \cdot \cos(\omega \cdot x_i)$$
$$n = 2; A = 10; \omega = \pi; x_i \in [-5.12, 5.12]$$

Figure 5-17 shows the Rastrigin function with 36 peaks and 25 troughs over the given design space.

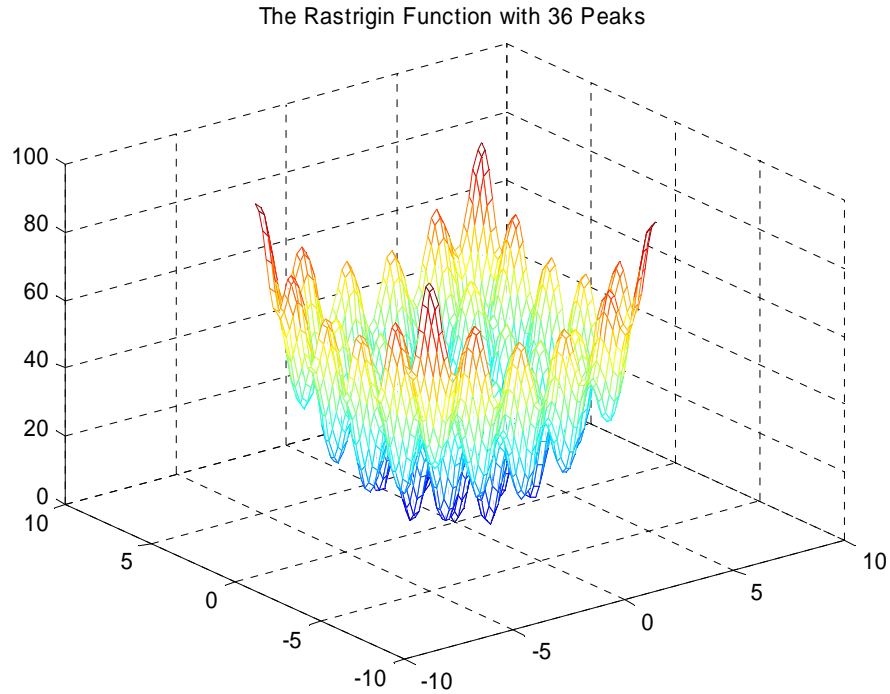


Figure 5-17: Illustration of the Rastrigin Function with 36 Peaks

The values of the general parameters and goodness of fit are listed in Table 6, and the visualization of the resulted surrogate models is shown in Figure 5-18 – Figure 5-24.

Table 6: Values of General Parameters and Goodness of Fit for the Rastrigin Function with 36 Peaks

Case	Method	Parameters of SVR	R^2	RMSE _{Tm}	RMSE _{RCV}	RMSE _{MPE}
1	RSM		0.5783	10.66359	5.70557	10.50619
	SVR	$C = 200; \varepsilon = 0.01; \sigma = 0.7$	1.0000	0.01000	12.76412	11.32161
	Hybrid	$C = 200; \varepsilon = 0.01; \sigma = 0.7$	1.0000	0.01000	7.38529	8.07982
2	RSM		0.5783	10.66359	7.64147	10.42822
	SVR	$C = 87.45; \varepsilon = 0.0166; \sigma = 0.7$	0.99811	0.71784	5.01745	8.38019
	Hybrid	$C = 32.12; \varepsilon = 0.0307; \sigma = 0.7$	0.98210	2.24653	3.97741	6.15362
3	RSM		0.5783	10.66359	7.60002	9.58444
	SVR	$C = 87.45; \varepsilon = 0.0166; \sigma = 0.057$	0.99999	0.06214	9.04092	12.14723
	Hybrid	$C = 32.12; \varepsilon = 0.0307; \sigma = 0.050$	1.00000	0.03055	4.34022	5.35881

RSM - Rastrigin Function with 36 Peaks

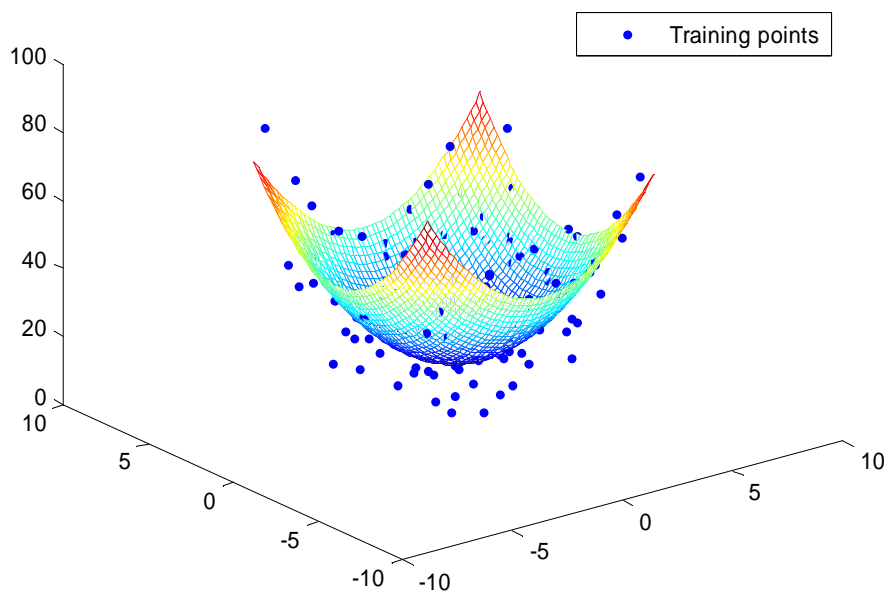


Figure 5-18: Surrogate Model for the Rastrigin Function by RSM – Case 1, 2, 3

SVR - Rastrigin Function with 36 Peaks

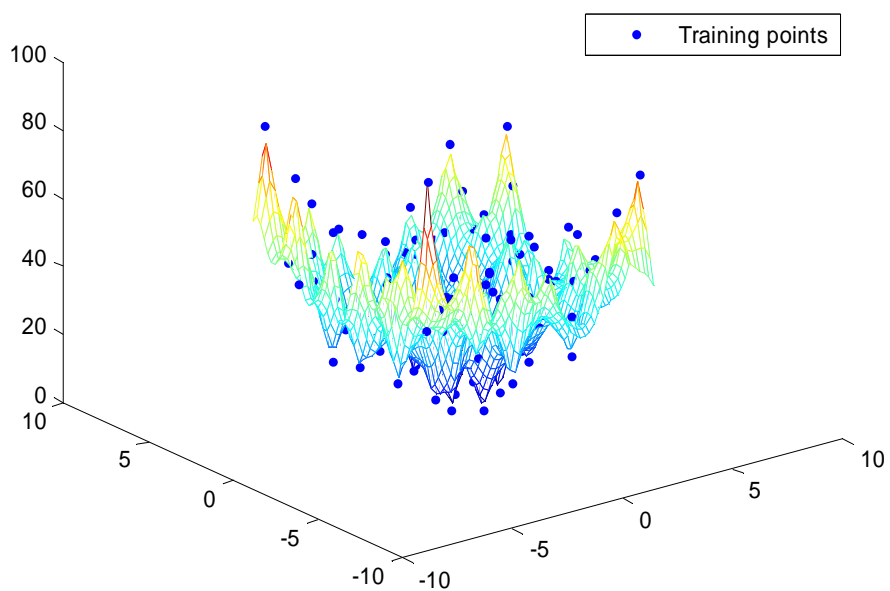


Figure 5-19: Surrogate Model for the Rastrigin Function by SVR – Case 1

Hybrid - Rastrigin Function with 36 Peaks

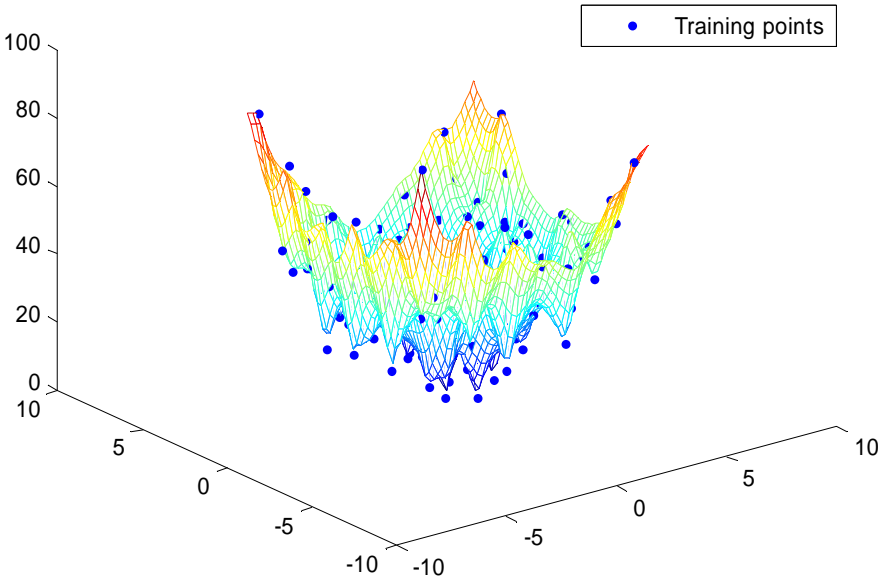


Figure 5-20: Surrogate Model for the Rastrigin Function by Hybrid – Case 1

SVR - Rastrigin Function with 36 Peaks

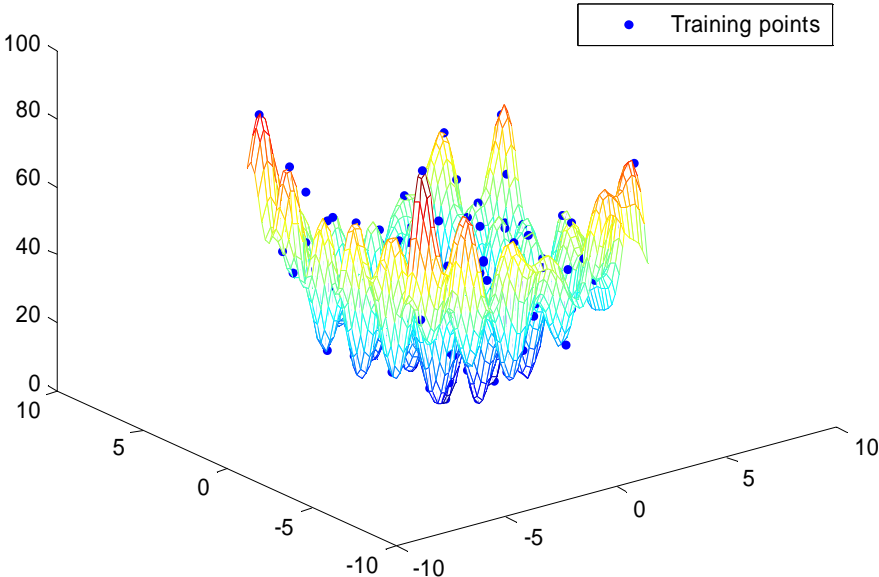


Figure 5-21: Surrogate Model for the Rastrigin Function by SVR – Case 2

Hybrid - Rastrigin Function with 36 Peaks

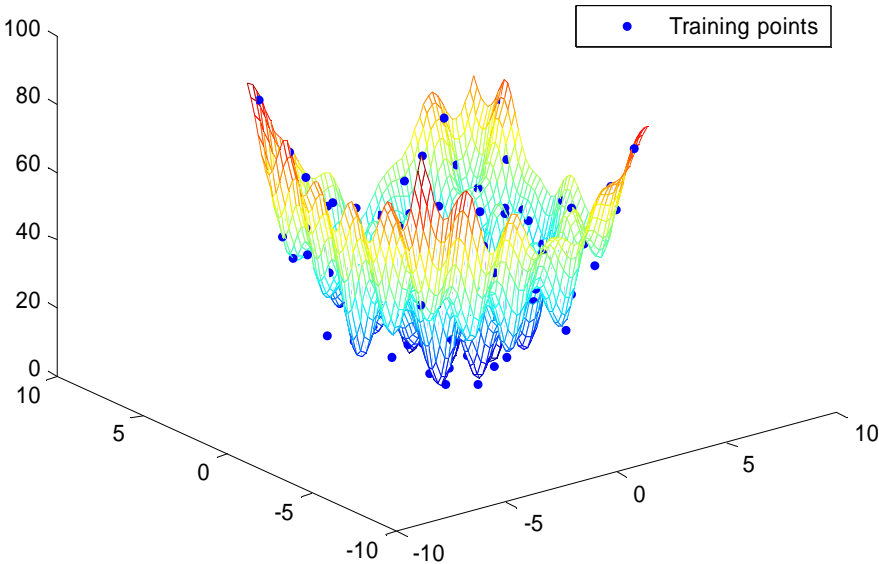


Figure 5-22: Surrogate Model for the Rastrigin Function by Hybrid – Case 2

SVR - Rastrigin Function with 36 Peaks

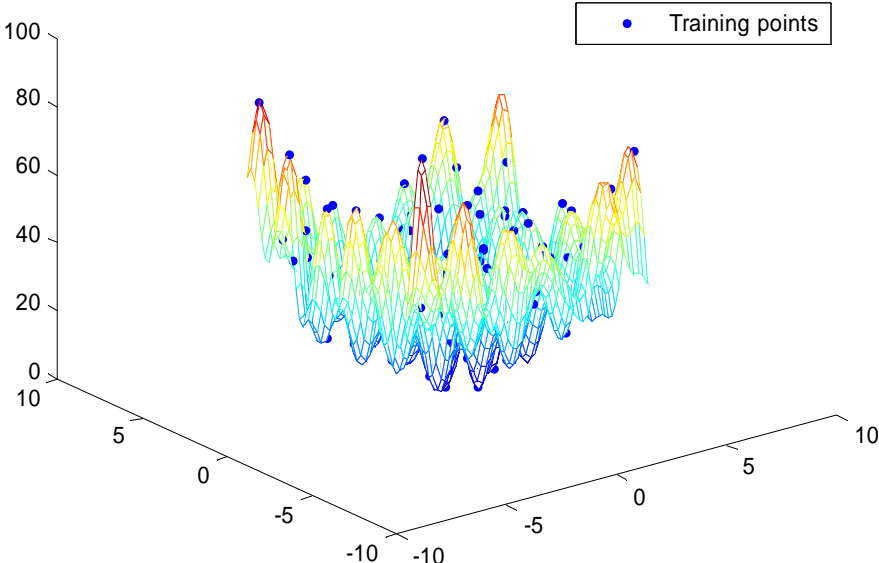


Figure 5-23: Surrogate Model for the Rastrigin Function by SVR – Case 3

Hybrid - Rastrigin Function with 36 Peaks

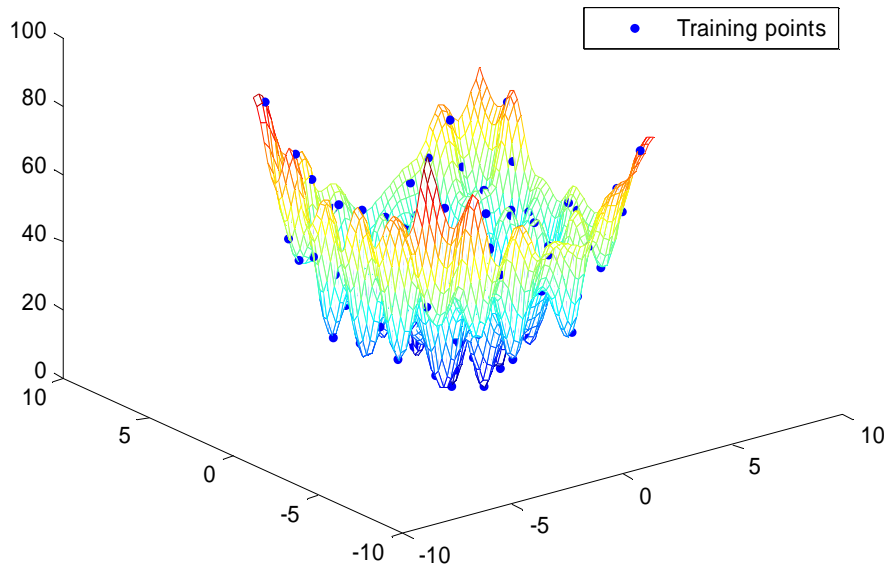


Figure 5-24: Surrogate Model for the Rastrigin Function by Hybrid – Case 3

5.1.4 Comparison with Neural Network

In Ref. [35], SVR is compared with RSM, MARS, RBF, and KG, but ANN is not compared with. Now that an integrated Neural Network software package BRAINN [101] is available that can automatically determine the number of hidden layer nodes and prevent overfitting in many cases, the hybrid method is compared with ANN.

To have a fair comparison, the following setup has been established. First, the same training sample is fed into the hybrid method and ANN. For the hybrid method, the whole sample is used for model construction, while for ANN, 80% of the sample points are used for model construction, and the other 20% points are used as validation cases to determine the best number of hidden layer nodes, which is also a part of the surrogate model. Second, the same random sample is used for MPE calculation. Third, the algorithm to choose the best value of σ , the third pre-specified parameter of SVR, is

changed to minimizing the following combined error ‘comb_error’ in Equation 5.1, instead of minimizing a modified information criterion. This is because the ANN model obtained with the BRAINN package minimizes both MFE and MPE, instead of a modified information criterion. By minimizing ‘comb_error’, a hybrid method model can also minimize both MFE and MPE.

$$\text{comb_error} = \ln(0.5 * \hat{\sigma}_{MLE}^2 + 0.5 * \text{RMSE}_{RCV}^2) \quad (5.1)$$

The Upper Hemisphere, Wave function, and Rastrigin function with 36 peaks are used again. For the Upper Hemisphere example, 100 sampling points are used; for the other two examples, 120 points are used. The values of the general parameters and goodness of fit are listed in Table 7. In this table, the column RMSE_{Tm} is the model fitting error, and RMSE_{MPE} is the true model predicting error calculated with random samples. The visualization of the resulted surrogate models is shown in Figure 5-25 – Figure 5-30.

Table 7: Goodness of Fit of RSSVR and NN for Three Pure Mathematical Examples

Example	Method	Number of hidden layer nodes Parameters of SVR	R^2	RMSE_{Tm}	RMSE_{MPE}
Upper Hemisphere	ANN	55	0.99375	0.16737	0.34982
	Hybrid	$C = 1.487; \varepsilon = 0.020; \sigma = 0.106$	0.99522	0.17708	0.31499
Wave function	ANN	15	0.99988	0.27333	0.88934
	Hybrid	$C = 76.14; \varepsilon = 0.012; \sigma = 0.097$	0.99968	0.61234	2.99815
Rastrigin function	ANN	20	0.70044	6.97226	11.32308
	Hybrid	$C = 32.12; \varepsilon = 0.031; \sigma = 0.057$	0.99934	0.42737	6.05625

ANN - Upper Hemisphere

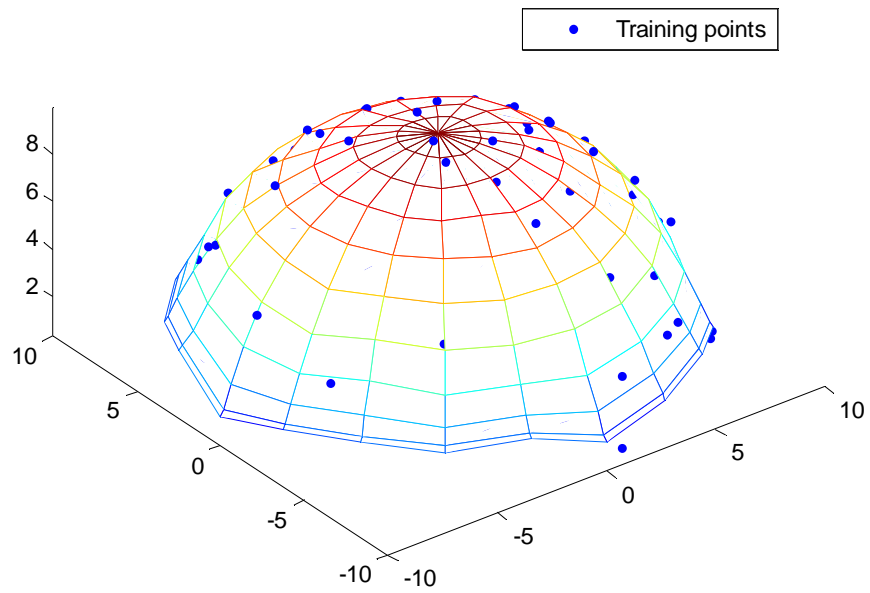


Figure 5-25: Surrogate Model for the Upper Hemisphere by ANN – Comparison

Hybrid - Upper Hemisphere

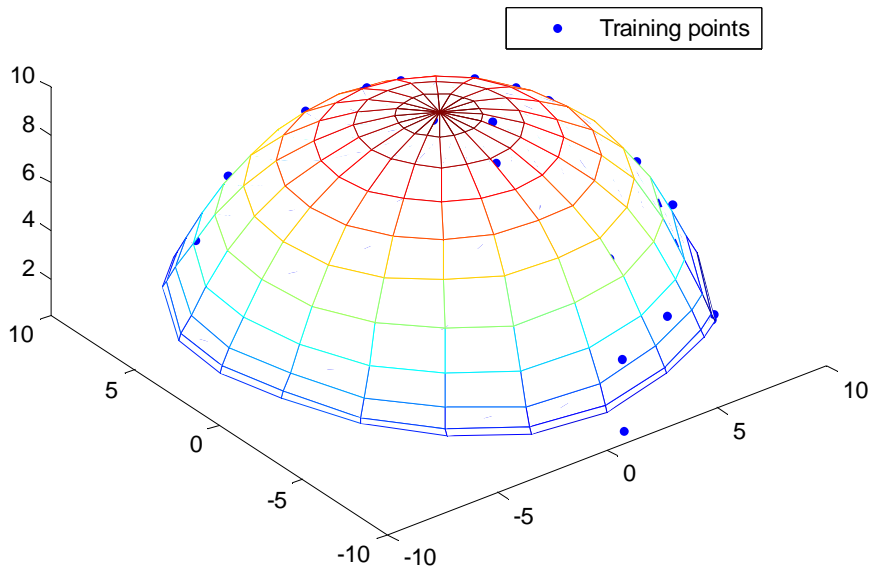


Figure 5-26: Surrogate Model for the Upper Hemisphere by Hybrid – Comparison

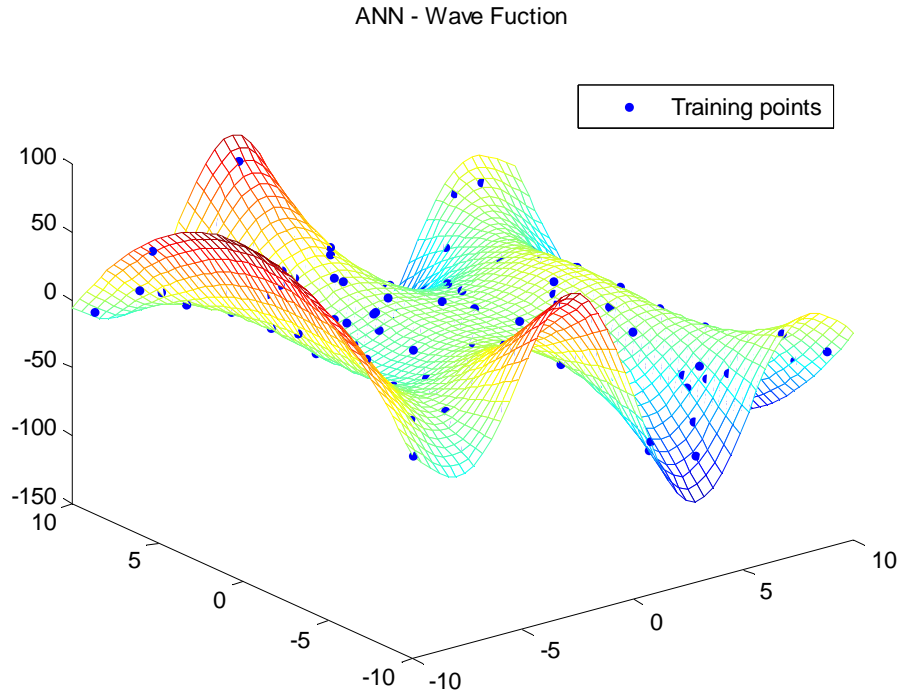


Figure 5-27: Surrogate Model for the Wave Function by ANN – Comparison

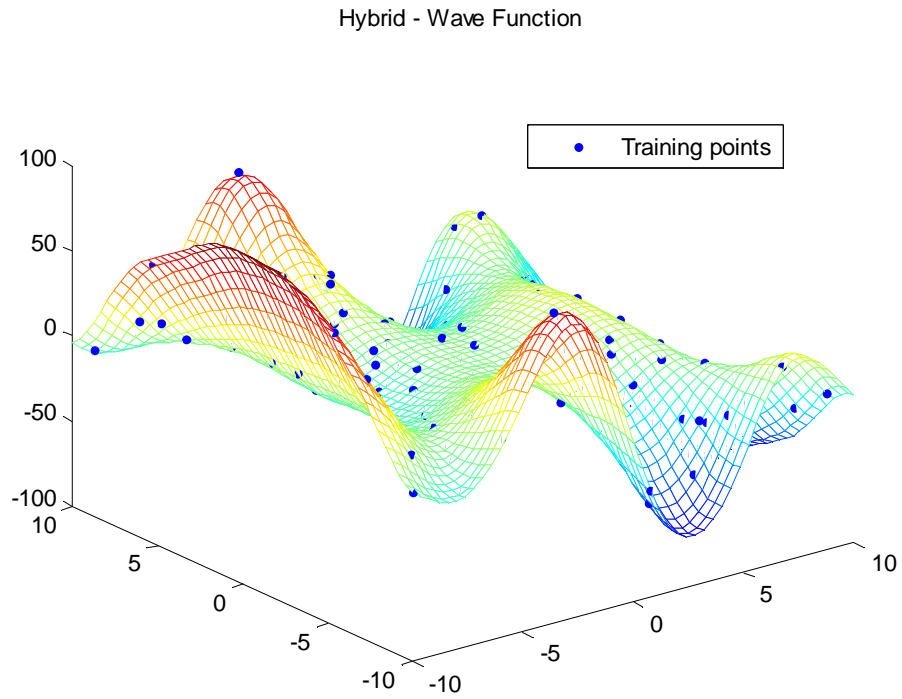


Figure 5-28: Surrogate Model for the Wave Function by Hybrid – Comparison

ANN - Rastrigin Function with 36 Peaks

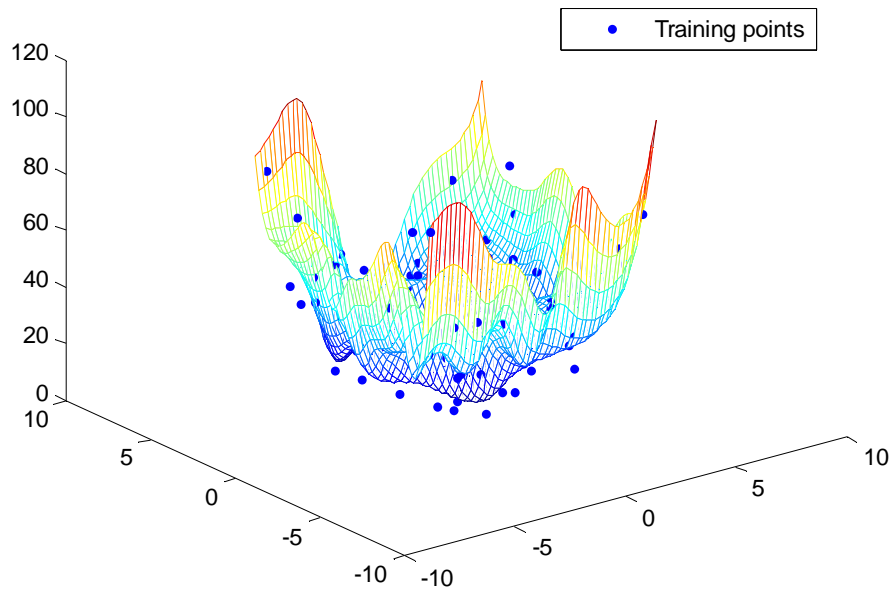


Figure 5-29: Surrogate Model for the Rastrigin Function by ANN – Comparison

Hybrid - Rastrigin Function with 36 Peaks

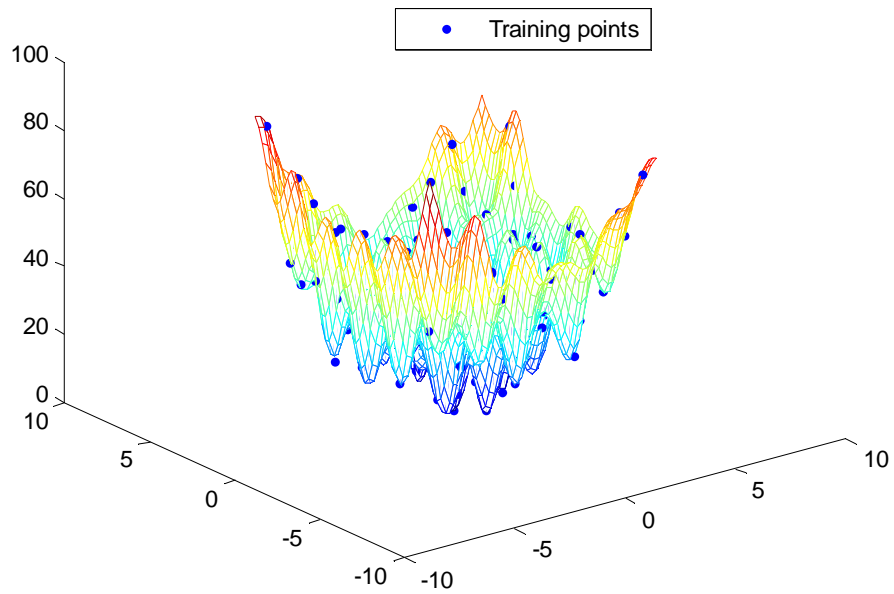


Figure 5-30: Surrogate Model for the Rastrigin Function by Hybrid – Comparison

5.1.5 Discussion

In Table 4 – Table 7, the column $RMSE_{Tm}$ is the model fitting error, $RMSE_{RCV}$ is the estimated model predicting error using the Random Cross Validation method, and $RMSE_{MPE}$ is the true model predicting error calculated with random samples. Observing and analyzing the model fitting results of the three pure mathematical examples, the following conclusions can be drawn:

1. Being used individually, RSM is good for low nonlinear problems such as the example of upper hemisphere, while SVR and hybrid method are good for both low and high nonlinear problems such as the examples of wave function and Rastrigin function.

2. The surrogate models constructed by the hybrid method almost always have the best accuracy in terms of both MFE and MPE, with different kernel functions and methods to select the parameters of SVR. This is because RSM can well capture the global behavior of the problem, while SVR can well capture the local nonlinear behavior.

3. With increase of complexity of the problems, the differences of accuracy between the surrogate models constructed by the hybrid method and the others become more substantial, especially when the parameters C , ε , and σ of SVR are automatically selected. This can be seen in the Case 3 of the example of the Rastrigin function. In this case, the hybrid method recovers two more peaks in the upper middle part than SVR.

4. The ERBF kernel is very good for MFE, i.e. the MFE of the surrogate models constructed with it is very small, but not for MPE. On the other hand, the GRBF kernel is good for both MFE and MPE, and works well for all examples.

Remember that for SVR and the hybrid method, Case 1 uses ERBF kernel, and Cases 2 and 3 use GRBF kernel. In Table 4 – Table 6, one can see that the MFE's of

Case 1 are always the smallest in the three cases, but in most cases the MPE's of Case 1 are the greatest; the MFE's of Cases 2 and 3 are small, although not as good as those of Case 1; the MPE's of Cases 2 and 3 are small and usually better than those of Case 1.

5. The random cross validation method can provide reasonable estimation for the MPE, especially for the surrogate models constructed by the hybrid method. The $RMSE_{RCV}$ is quite close to the $RMSE_{MPE}$ as can be seen in all examples.

6. The automatic process for selection of parameters of SVR works very well, since the MFE's and MPE's of all the automatically fitted SVR and hybrid models are small. This includes normalization of the values of the design variables, selection of the parameters C and ε by the practical method, selection of the kernel parameter σ by minimizing a modified information criterion. Thus the users do not need to select the pre-specified parameters now and can obtain good results like an expert of SVR.

7. With a small sample size, i.e. 120, the SVR and hybrid methods obtained satisfactory results for the complex examples of wave function and Rastrigin function. The MFE's are less than 1% and MPE's less than 5%. Other examples that are not provided here show that the accuracy of the surrogate models constructed by RSM increases little or does not with the sample size once the order of the polynomials is selected, while for SVR and hybrid it steadily increases with the sample size. All of these confirm that the mathematical foundation of SVR overcomes the problem of "curse of dimensionality".

8. With a small sample size of 100 or 120, the hybrid method can obtain accurate models for many types of complex problems, while the ANN method can not. From Table 7 and Figure 5-25 – Figure 5-30, one can see that the hybrid method obtains high

accuracy for all the three examples in terms of MFE and MPE; the ANN obtains high accuracy for the examples of Upper Hemisphere and Wave function, but low accuracy for the example of Rastrigin function. Especially, only a few peaks and troughs are recovered with ANN, and this makes the error MPE is very high.

5.2 Pure Mathematical Examples of Finding the Weak Pareto Frontier

Three two-objective and one three-objective deterministic optimization problems are used to demonstrate that this framework can surely find the exact weak Pareto frontier. Those examples are selected in this research because those examples have different features, such as multiple-to-one mapping from the design space to the objective space, frontier of disjointed segments, and more than two objectives with a constraint. Table 8 lists the objective functions and features of these four examples. All these four examples have no more than three objectives and thus can be visualized. For each example, only the second search scheme is used since the first search scheme can not be used without coupling variables.

Table 8: Objective Functions and Features of the Mathematical Examples of Finding WPF

	Functions	Feature
P1	$f_1 = x_1$ $f_2 = a / x_1$ $a = 2 - \exp\left(\frac{-(x_2 - 0.2)^2}{0.08}\right) - 0.8 \exp\left(\frac{-(x_2 - 0.6)^2}{0.4}\right)$ $0.002 \leq x_1 \leq 0.2$ $0 \leq x_2 \leq 1$	Uniform sampling in the design space corresponds to nonuniform distribution in the objective space

P2	$f_1 = \cos(a) * b$ $f_2 = \sin(a) * b$ $a = \frac{\pi}{180} (45 + 40 \sin(2\pi x_1) + 25 \sin(2\pi x_2))$ $b = 1 + 0.5 \cos(2\pi x_2)$ $0 \leq x_1 \leq 1$ $0 \leq x_2 \leq 1$	<p>Compound frontier of convex and concave segments;</p> <p>Multiple-to-one mapping from the design space to the objective space</p>
P3	$f_1 = x_1$ $f_2 = a * b$ $a = 1 + 10x_2$ $b = 1 - \left(\frac{x_1}{a}\right)^2 - \frac{x_1}{a} \sin(8\pi x_1)$ $0 \leq x_1 \leq 1$ $0 \leq x_2 \leq 1$	<p>Frontier of disjointed segments</p>
P4	$f_1 = 25 - \frac{x_1^3 + x_1^2(1 + x_2 + x_3) + x_2^3 + x_3^3}{10}$ $f_2 = 35 - \frac{x_1^3 + 2x_2^3 + x_2^2(2 + x_1 + x_3 + x_3^3)}{10}$ $f_3 = 50 - \frac{x_1^3 + x_2^3 + 3x_3^3 + x_3^2(3 + x_1 + x_2)}{10}$ $g_1 = 12 - x_1^2 + x_2^2 + x_3^2 \leq 0$ $0 \leq x_1 \leq 10$ $0 \leq x_2 \leq 10$ $0 \leq x_3 \leq 10$	<p>High dimensional example: three objectives;</p> <p>Nonsolid objective space: intertwined three dimensional surface;</p> <p>Uniform sampling in the design space corresponds to nonuniform distribution in the objective space;</p> <p>One constraint</p>

5.2.1 First Mathematical Example of Finding WPF

For this example, the size of S_2 or the number of search starting points is given the same as $S_0 = 4950$ with 99% probability and 2% error, since this problem is simple and thus less search starting points are needed. The objective functions and the feature are given as following:

	Functions	Feature
P1	$f_1 = x_1$ $f_2 = a / x_1$ $a = 2 - \exp\left(\frac{-(x_2 - 0.2)^2}{0.08}\right) - 0.8 \exp\left(\frac{-(x_2 - 0.6)^2}{0.4}\right)$ $0.002 \leq x_1 \leq 0.2$ $0 \leq x_2 \leq 1$	Uniform sampling in the design space corresponds to nonuniform distribution in the objective space

The shape of the objective space is designated in Figure 5-31 by the area enclosed by the blue curves. The true weak Pareto frontier is the curve closer to the origin, including a vertical segment and an almost horizontal segment.

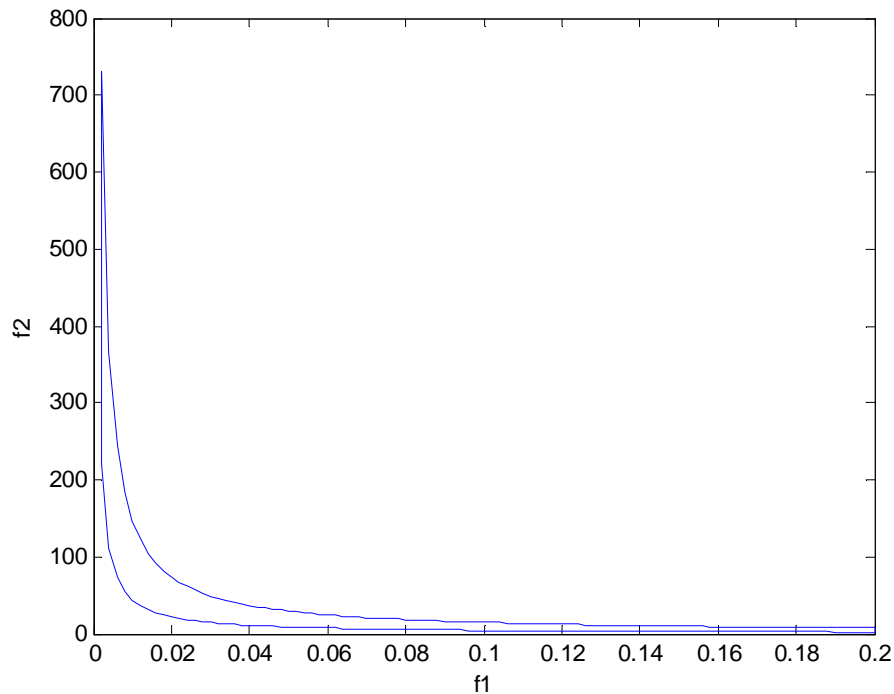


Figure 5-31: Objective Space of the First Mathematical Example of Find WPF

The results are given in Figure 5-32. From this figure, one can see the WPF points obtained by the second search scheme distributed along the true WPF. Comparing the left

side with the right side of this figure, one can see that because the magnitude level of the second objective function is quite greater than that of the first objective function, the uniform sampling in the design space corresponds to nonuniform distribution of points in the objective space. Even though, many WPF points are obtained along the left vertical segment of the true WPF and the distribution of these points is roughly even.

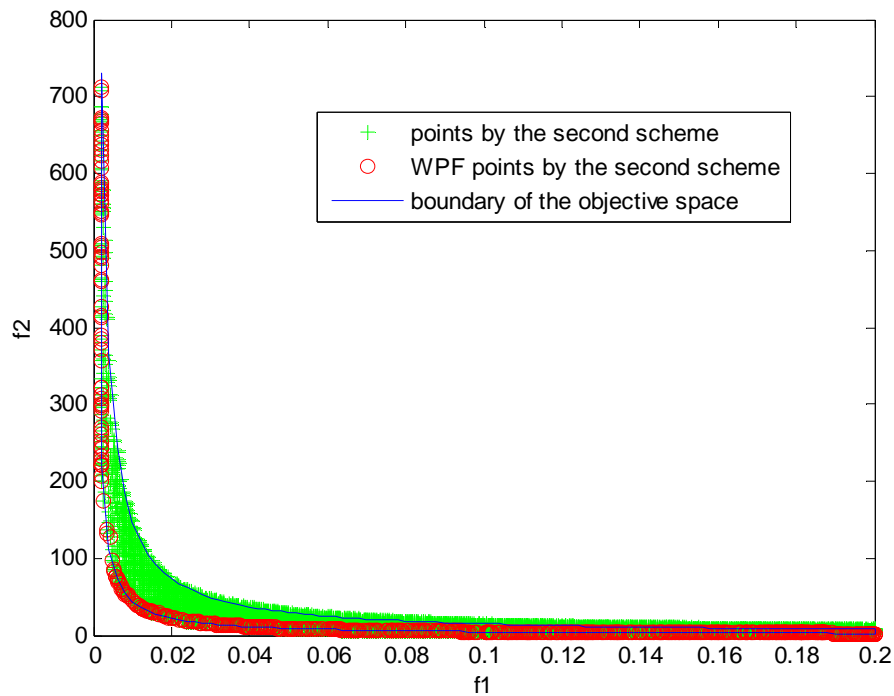


Figure 5-32: Results of the First Mathematical Example of Finding WPF

5.2.2 Second Mathematical Example of Finding WPF

For this example, the size of S_2 or the number of search starting points is given the same as $S_0 = 4950$ with 99% probability and 2% error, since this problem is simple and thus less search starting points are needed. The objective functions and the feature are given as following:

	Functions	Feature
P2	$f_1 = \cos(a) * b$ $f_2 = \sin(a) * b$ $a = \frac{\pi}{180} (45 + 40 \sin(2\pi x_1) + 25 \sin(2\pi x_2))$ $b = 1 + 0.5 \cos(2\pi x_2)$ $0 \leq x_1 \leq 1$ $0 \leq x_2 \leq 1$	Compound frontier of convex and concave segments; Multiple-to-one mapping from the design space to the objective space

The shape of the objective space is designated in Figure 5-33 by the shape represented by the blue points. The true weak Pareto frontier is the edges closer to the origin, including one concave segment and two convex segments. Also one can find that in some areas the blue points are much denser. This phenomenon means that the mapping from the design space to the objective space is multiple-to-one.

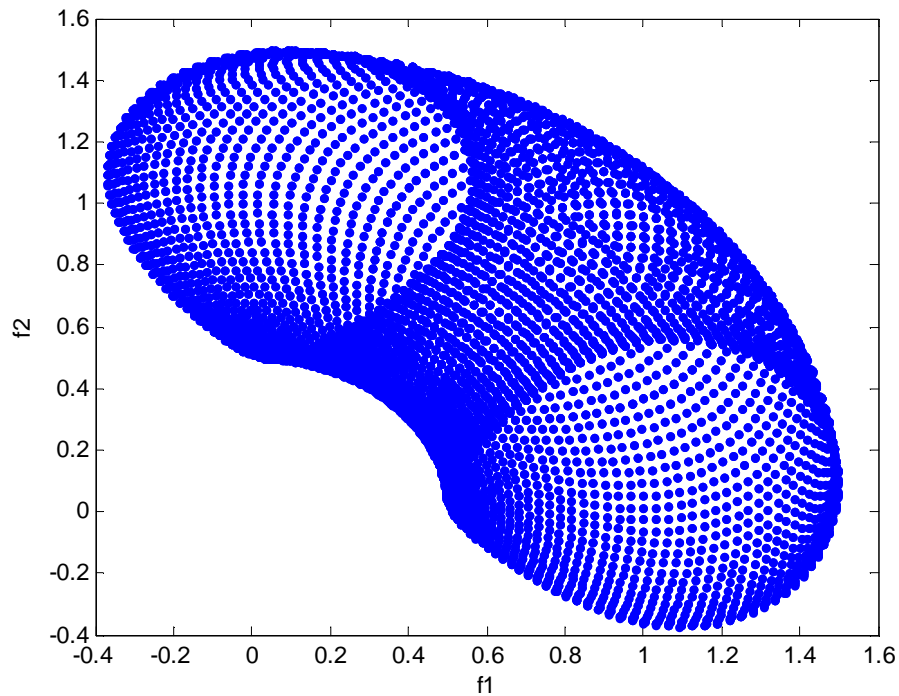


Figure 5-33: Objective Space of the Second Mathematical Example of Find WPF

The results are given in Figure 5-34. From this figure, one can see the WPF points obtained by the second search scheme evenly distributed along the true WPF, and the number of WPF points is plentiful.

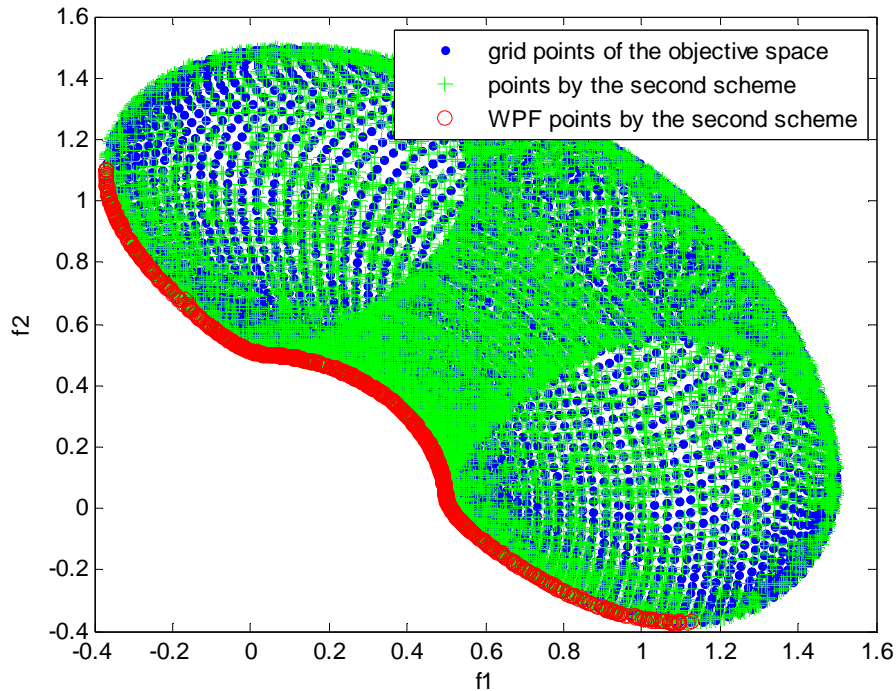


Figure 5-34: Results of the Second Mathematical Example of Finding WPF

5.2.3 Third Mathematical Example of Finding WPF

For this example, the size of S_2 or the number of search starting points is given the same as $S_0 = 4950$ with 99% probability and 2% error, since this problem is simple and thus less search starting points are needed. The objective functions and the feature are given as following:

	Functions	Feature
P3	$f_1 = x_1$ $f_2 = a * b$ $a = 1 + 10x_2$ $b = 1 - \left(\frac{x_1}{a}\right)^2 - \frac{x_1}{a} \sin(8\pi x_1)$ $0 \leq x_1 \leq 1$ $0 \leq x_2 \leq 1$	Frontier of disjointed segments

The shape of the objective space is designated in Figure 5-35 by the area enclosed by the blue curves. Because of the wavelike curve at the bottom of the objective space, the true weak Pareto frontier consists of spatially disjointed segments, including a vertical segment at the left of the objective space.

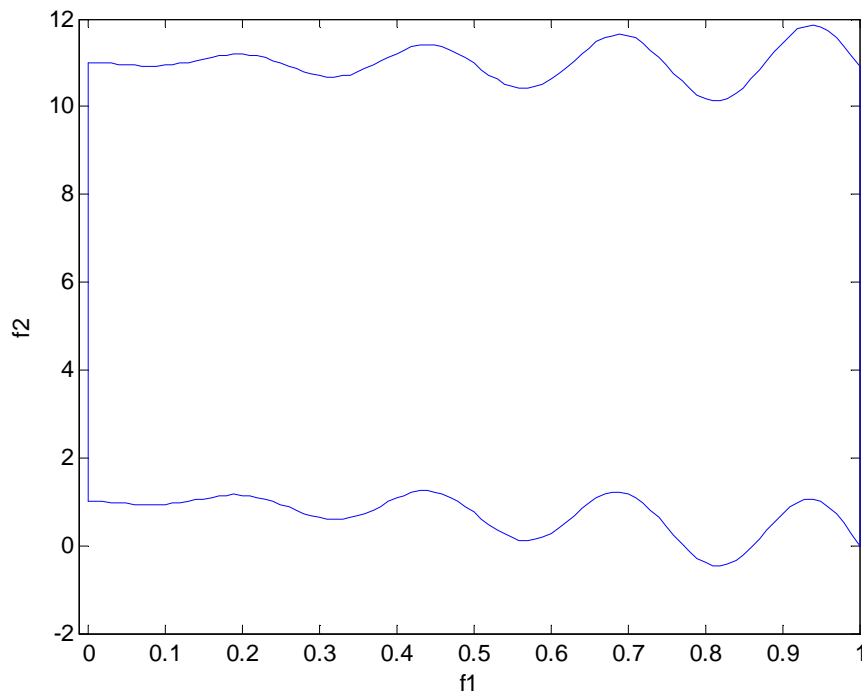


Figure 5-35: Objective Space of the Third Mathematical Example of Find WPF

The results are given in Figure 5-36. From this figure, one can see the WPF points obtained by the second search scheme evenly distributed along the true WPF consisting of spatially disjointed segments, and the number of WPF points is plentiful.

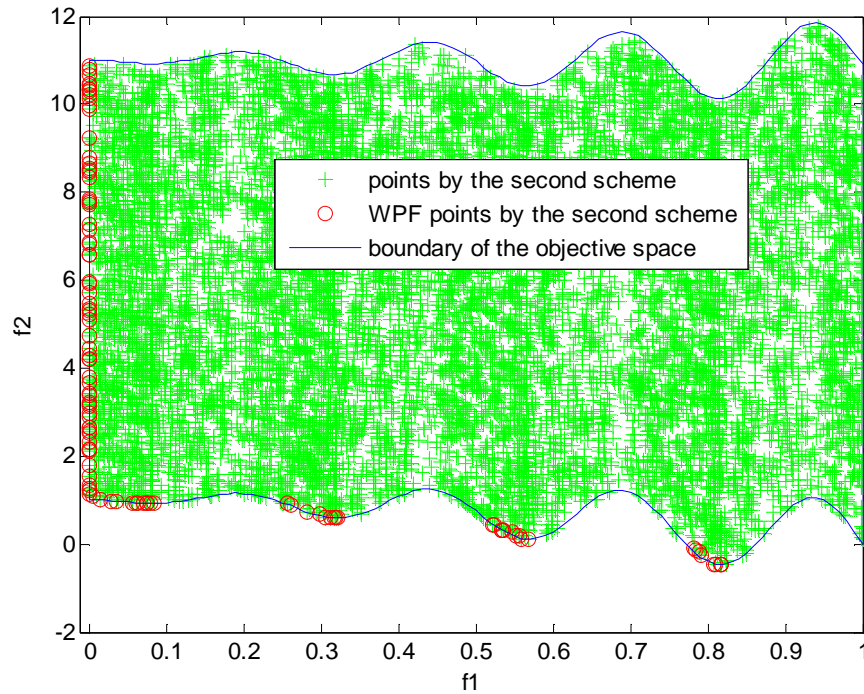


Figure 5-36: Results of the Third Mathematical Example of Finding WPF

5.2.4 Fourth Mathematical Example of Finding WPF

This example is close to a real engineering problem because it has three design variables, three objectives, and one constraint, although a real engineering problem usually has more design variables, objectives, and constraints.

For this example, the size of S_2 or the number of search starting points is given the same as $S_0 = 4950$ with 99% probability and 2% error, since this problem is simple and

thus less search starting points are needed. The objective functions and the feature are given in the table below.

The shape of the objective space is designated in Figure 5-37 by the shape represented by the blue points. Actually, the shape of objective space is not easy to imagine, but it is not solid but includes intertwined three dimensional surfaces. The true weak Pareto frontier is also not easy to see, but it is a three dimensional curve.

	Functions	Feature
P4	$f_1 = 25 - \frac{x_1^3 + x_1^2(1 + x_2 + x_3) + x_2^3 + x_3^3}{10}$ $f_2 = 35 - \frac{x_1^3 + 2x_2^3 + x_2^2(2 + x_1 + x_3 + x_3^3)}{10}$ $f_3 = 50 - \frac{x_1^3 + x_2^3 + 3x_3^3 + x_3^2(3 + x_1 + x_2)}{10}$ $g_1 = 12 - x_1^2 + x_2^2 + x_3^2 \leq 0$ $0 \leq x_1 \leq 10$ $0 \leq x_2 \leq 10$ $0 \leq x_3 \leq 10$	<p>High dimensional example: three objectives;</p> <p>Nonsolid objective space: intertwined three dimensional surfaces;</p> <p>One constraint</p>

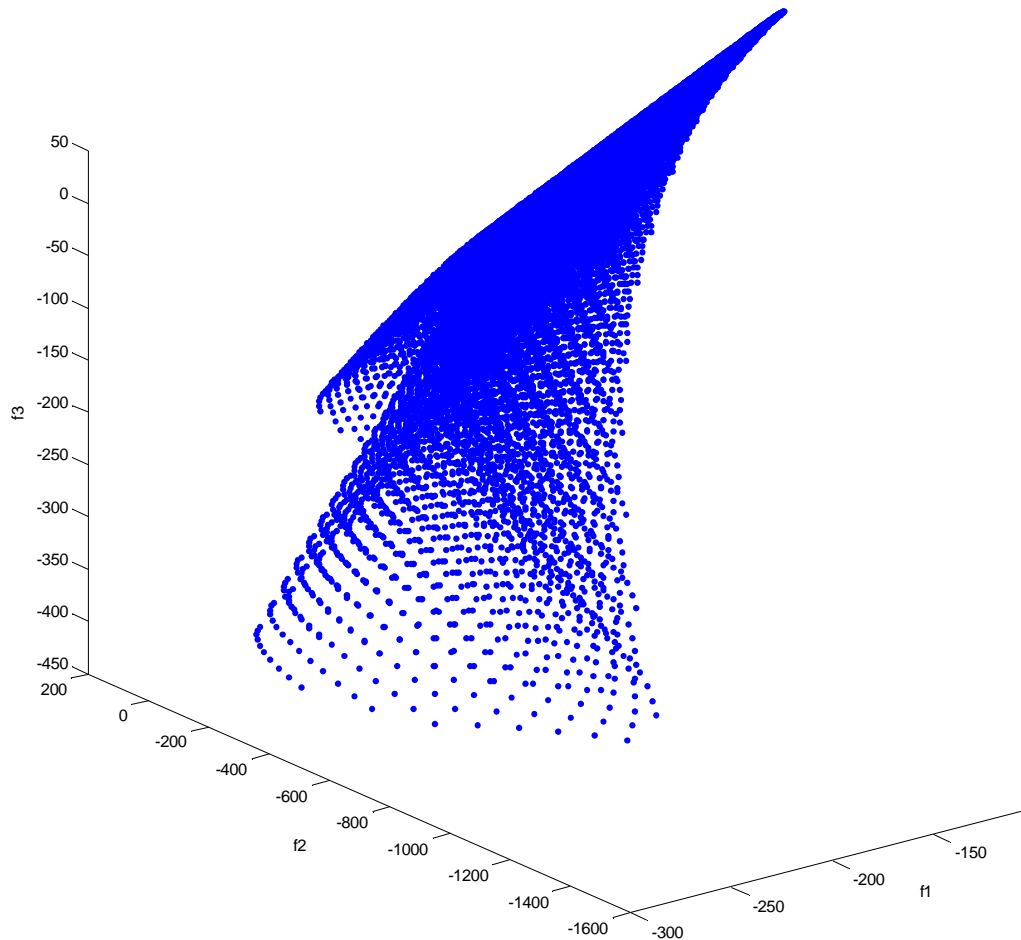


Figure 5-37: Objective Space of the Fourth Mathematical Example of Find WPF

The results are given in Figure 5-38. From this figure, one can see the WPF points obtained by the second search scheme evenly distributed along the true WPF, and the number of WPF points is plentiful. Also one can see the true PF and WPF are quite different, and as mentioned previously, the WPF has more solutions besides the PF solutions.

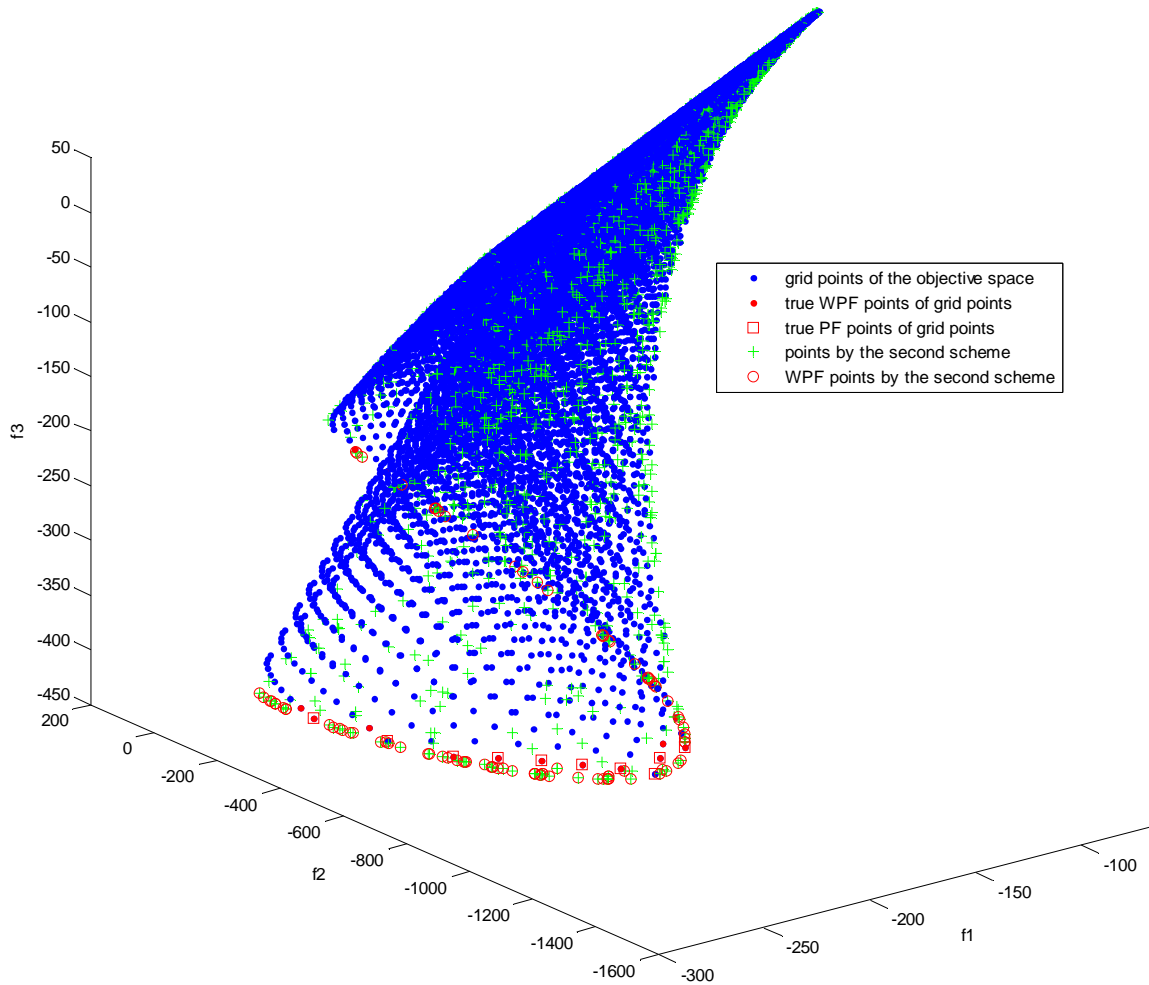


Figure 5-38: Results of the Fourth Mathematical Example of Finding WPF

5.2.5 Discussion

Although only the second search scheme is used for each example since the first search scheme can not be used without coupling variables, observing and analyzing the results of finding WPF for the four pure mathematical examples, the following conclusions can be drawn:

1. The second search scheme for finding WPF is a generic for many types of problems, i.e. whatever the features and the complexity of the problems are, and it can surely find the true WPF with exactly the same procedures.

2. The second search scheme for finding WPF can find not only the local WPF, but also the global WPF. It will not be trapped by the local WPF. This can be seen in the third example of which the WPF comprises 4 disjointed segments (local WPF segments).

3. The second search scheme for finding WPF can find a large number of WPF points even with smaller number of search starting points than that estimated with the rule-of-thumb equation given in this research. This can be seen in all four examples.

4. The WPF points found by the second search scheme are nearly evenly distributed over the complete frontier, or nearly evenly distributed over each of the segments of the complete frontier even if uniform sampling in the design space corresponds to nonuniform distribution in the objective space. This can be seen in all four examples.

5.3 A Transport Aircraft Design Example

A simple yet typical aircraft design problem is used to show the feasibility of the new framework of determination of the WPF solutions under probabilistic constraints. Here ‘simple’ just means the disciplinary analyses are formulated with explicit equations, which do not exist in a real design process. This problem has

1) 4 disciplinary analyses;

2) 7 system level design variables including 2 coupling variables that are assumed to be normally distributed about the mean values with 3σ symmetrical truncation (see APPENDIX G for a summary of the doubly-truncated normal distribution);

3) 7 PC’s all with required POS of 0.85; and

4) 2 design objectives. See APPENDIX H for detailed information. Figure 2-21 is repeated here to show the DSM of this problem.

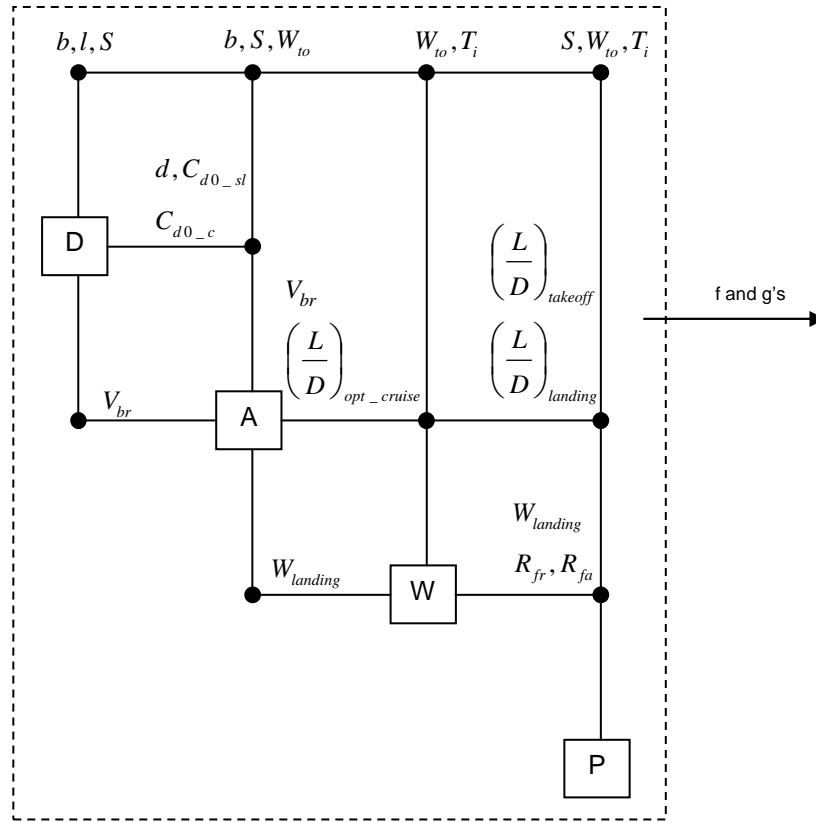


Figure 2-21: Example of the DSM of a Multidisciplinary Aircraft Design Problem

The surrogate models are first constructed, and then those models are used to find the WPF and its design solutions. For the purpose of validation, the original CA's are also used to find the WPF, and the two kinds of WPF's obtained with the surrogate models and original CA's are compared. The exact single-objective deterministic optimal solutions and objective values over the given design space are obtained and given in Table 9. Here 'exact' implicates the original CA's are used.

Table 9: Exact Single-Objective Deterministic Optimal Results of the Aircraft Design Example

Objective	Objective Value	b	l	S	W_{to}	T_i
PI	151.11	111.76	130.43	1429.31	163567.1	27746.8
T_i	20765.66	139.37	135.36	1850.00	160108.8	20765.66

5.3.1 Surrogate Models

For this example, discipline-level surrogate models are constructed. A surrogate model is constructed for each of the responses of the discipline analyses. For each response, the training sample for surrogate model construction includes 150 points by HS sampling, the sample for estimation of true MPE includes 300 points by LHC sampling, and the sample for RCV includes 200 points by LHC sampling. The values of both the design variables and the response are normalized, the kernel is GRBF, the parameters C and ε are estimated by the practical estimation method, the parameter σ is selected by minimizing the modified information criterion BICC, and the best surrogate model is selected by minimizing the modified information criterion BICC as well.

The selected surrogate modeling methods and goodness of fit for the responses are listed in Table 10 – Table 13, where $\overline{\text{RMSE}}_{\text{Tm}}$ is the normalized model fitting error, $\overline{\text{RMSE}}_{\text{RCV,Hybrid}}$ is the normalized estimation of model predicting error using Random Cross Validation, $\overline{\text{RMSE}}_{\text{MPE}}$ is the normalized true model predicting error calculated with random samples, and RMSE_{RCV} is the (real) estimated model predicting error after de-normalization. The normalized values are actually percentage values since all responses are normalized to $[0, 100]$. The accuracy of the surrogate models is satisfactory, since the maximum normalized model predicting error is less than 3%, and most of errors are less than 1%. The results once again show that $\overline{\text{RMSE}}_{\text{RCV,Hybrid}}$ can provide reasonable estimation for $\overline{\text{RMSE}}_{\text{MPE}}$, i.e. the RCV method can provide reasonable estimation for the model predicting error.

Table 10: Values of General Parameters and Goodness of Fit for the D CA

Response	Method	$\overline{\text{RMSE}}_{\text{Tm}}$	$\overline{\text{RMSE}}_{\text{MPE}}$	$\overline{\text{RMSE}}_{\text{RCV,Hybrid}}$	RMSE_{RCV}
d	RSM	0.0443	0.0457	0.0321	5.619E-4
c_{d0_sl}	Hybrid	0.0076	0.2094	0.1983	5.758E-6
c_{d0_c}	Hybrid	0.0074	0.2030	0.1874	5.802E-6

Table 11: Values of General Parameters and Goodness of Fit for the A CA

Response	Method	$\overline{\text{RMSE}}_{\text{Tm}}$	$\overline{\text{RMSE}}_{\text{MPE}}$	$\overline{\text{RMSE}}_{\text{RCV,Hybrid}}$	RMSE_{RCV}
$\left(\frac{L}{D}\right)_{\text{takeoff}}$	RSM	0.1250	0.1525	0.1250	0.01282
$\left(\frac{L}{D}\right)_{\text{landing}}$	RSM	0.7373	1.0671	0.7373	0.10379
$\left(\frac{L}{D}\right)_{\text{opt_cruise}}$	RSM	0.0774	0.1295	0.0774	0.00838
V_{br}	RSM	0.1454	0.1773	0.1454	0.30913

Table 12: Values of General Parameters and Goodness of Fit for the W CA

Response	Method	$\overline{\text{RMSE}}_{\text{Tm}}$	$\overline{\text{RMSE}}_{\text{MPE}}$	$\overline{\text{RMSE}}_{\text{RCV,Hybrid}}$	RMSE_{RCV}
W_{landing}	Hybrid	0.0226	0.3842	0.3516	187.671
R_{fr}	Hybrid	0.0205	0.4921	0.4614	0.00112
R_{fa}	RSM	0.0321	0.0342	0.0201	0.00001
U	Hybrid	0.0201	0.4510	0.4225	0.00112

Table 13: Values of General Parameters and Goodness of Fit for the P CA

Response	Method	$\overline{\text{RMSE}}_{\text{Tm}}$	$\overline{\text{RMSE}}_{\text{MPE}}$	$\overline{\text{RMSE}}_{\text{RCV,Hybrid}}$	RMSE_{RCV}
S_{to}	Hybrid	0.0267	0.5663	0.3707	22.9246
S_l	Hybrid	0.0184	0.2960	0.1794	5.65700
q_{to}	Hybrid	0.0213	1.8260	1.2634	0.00201
q_l	Hybrid	0.0154	0.9421	0.5842	0.00148
\overline{R}_f	SVR	0.1667	2.5723	1.9700	0.03808

5.3.2 Design Results

For this design example, the new framework is implemented with the two neighborhood search schemes. For each search scheme with the surrogate models, two cases are considered: the first case uses sample sizes of S_2 as 9900, and the second case uses 19800. The number 19800 is the estimation given by Equation 4.13 with 99% probability and 2% error. For each valid solution of S_3 , the sample size of S_4 is given as 792 estimated by Equation 2.44 with 99% probability and 5% error.

With the first search scheme and surrogate models, 2518 and 4699 valid solutions S_3 are obtained, respectively. Then 549 and 2891 candidate points of S_5 are obtained, respectively. Finally, 14 and 41 WPF points are obtained, respectively. The figures of the WPF's in the objective space are shown in Figure 5-39 and Figure 5-40.

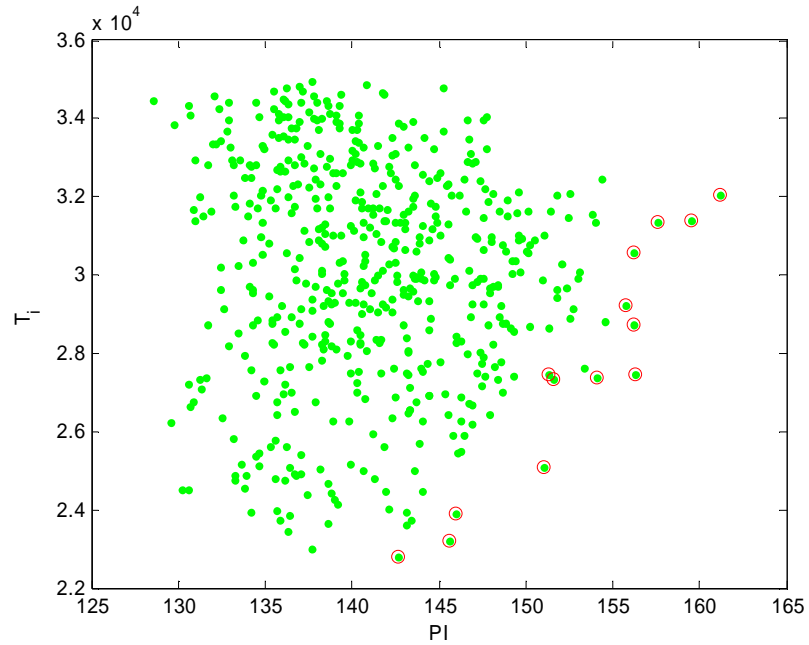


Figure 5-39: WPF Found by the First Search Scheme with 9900 points of S_2 and Surrogate Models for the Aircraft Design Example

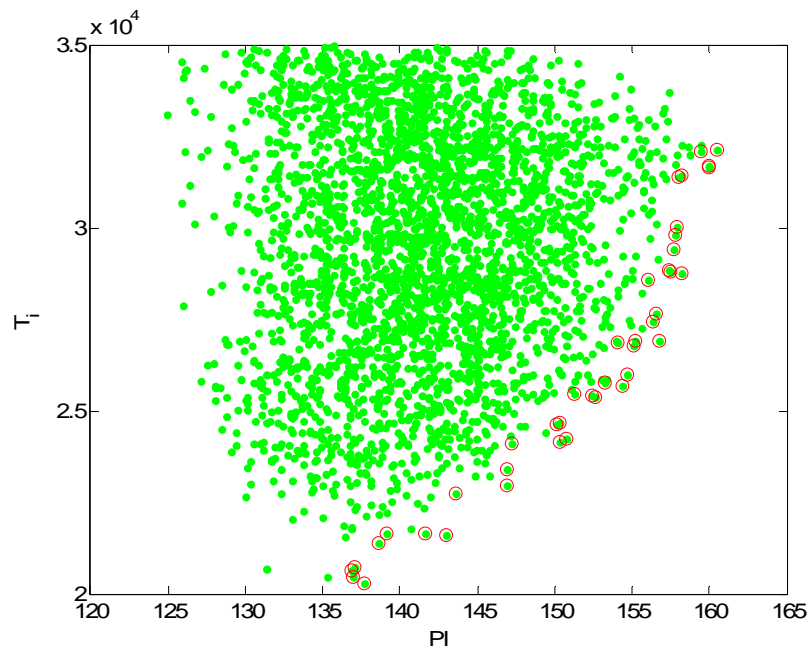


Figure 5-40: WPF Found by the First Search Scheme with 19800 points of S_2 and Surrogate Models for the Aircraft Design Example

With the first search scheme and surrogate models, the following valid solutions that are the closest to the single-objective deterministic optimal solutions are found and listed in Table 14. The distance used here to select the closest solution is the relative Euclidean distance.

Table 14: Valid Solutions Closest to Single-Objective Deterministic Optimal Solutions with the First Search Scheme and Surrogate Models for the Aircraft Design Example

Case	Objective	Objective Value	b	l	S	W_{to}	T_i
1	PI	151.28*, 151.06 ⁺	112.07	132.17	1457.06	159870.06	27463.25
	T_i	20750.27* ⁺	139.12	129.67	1758.82	162128.87	20750.27
2	PI	151.60*, 151.34 ⁺	110.66	131.08	1438.78	163948.09	27886.68
	T_i	20697.02* ⁺	138.86	132.20	1834.88	161679.72	20697.02

Note: *Predicted by the surrogate models;

⁺Predicted by the original CA's.

With the second search scheme and surrogate models, 4177 and 7235 valid solutions S_3 are obtained, respectively. Then 133 and 280 candidate points of S_5 are obtained, respectively. Finally, 11 and 10 WPF points are obtained, respectively. The figures of the WPF's are shown in Figure 5-41 and Figure 5-42.

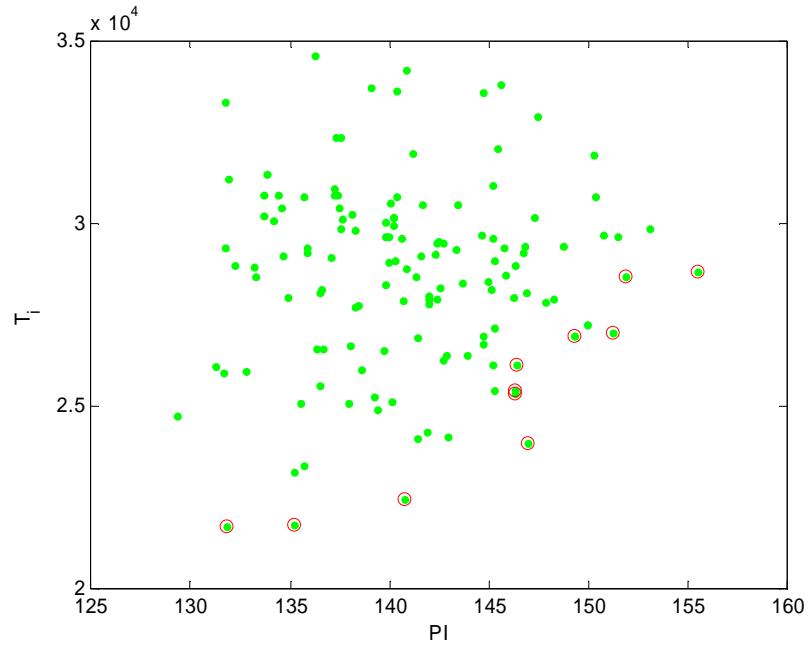


Figure 5-41: WPF Found by the Second Search Scheme with 9900 points of S_2 and Surrogate Models for the Aircraft Design Example

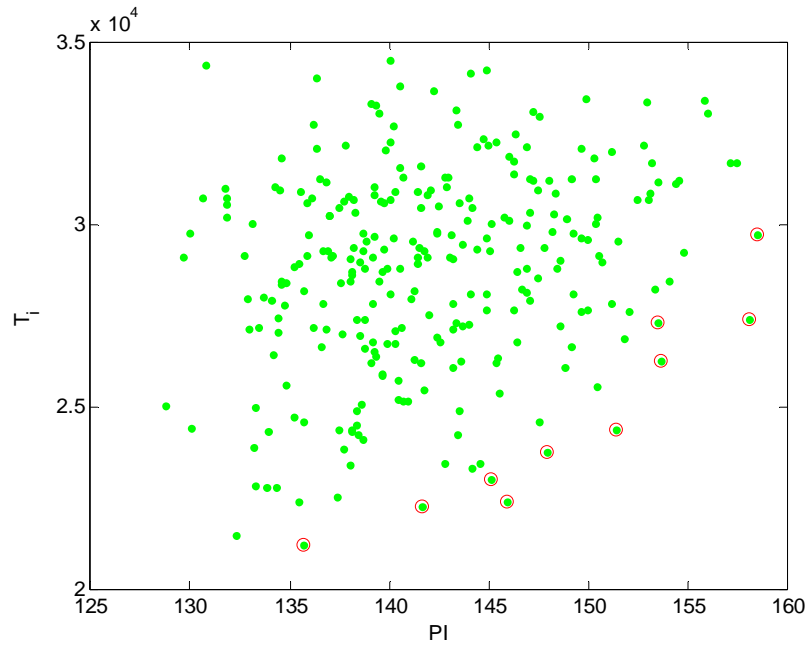


Figure 5-42: WPF Found by the Second Search Scheme with 19800 points of S_2 and Surrogate Models for the Aircraft Design Example

With the second search scheme and surrogate models, the following valid solutions that are the closest to the single-objective deterministic optimal solutions are found and listed in Table 15. The distance used here to select the closest solution is the relative Euclidean distance.

Table 15: Valid Solutions Closest to Single-Objective Deterministic Optimal Solutions with the Second Search Scheme and Surrogate models for the Aircraft Design Example

Case	Objective	Objective Value	b	l	S	W_{to}	T_i
1	PI	151.40*, 151.14 ⁺	111.96	129.51	1429.25	163465.15	27457.10
	T_i	20431.33* ⁺	139.10	138.05	1837.30	160187.87	20431.33
2	PI	150.50*, 150.23 ⁺	112.45	130.70	1438.69	164450.23	27865.69
	T_i	20671.82* ⁺	138.91	135.35	1832.20	161634.81	20671.82

Note: *Predicted by the surrogate models;

⁺Predicted by the original CA's.

The two WPF's found by the first search scheme with 9900 and 19800 points of S_2 respectively and surrogate models are compared in Figure 5-43; the two WPF's found by the second search scheme with 9900 and 19800 points of S_2 respectively and surrogate models are compared in Figure 5-44; and the two WPF's found by the two search schemes with 19800 points of S_2 and surrogate models are compared in Figure 5-45.

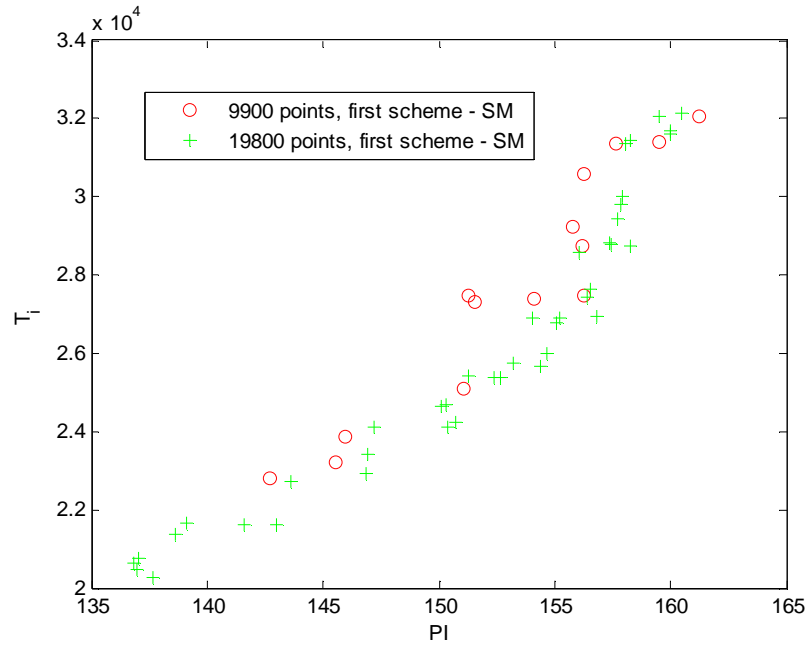


Figure 5-43: Comparison between the Two WPF's Found by the First Search Schemes with 9900 and 19800 Points of S_2 and Surrogate Models for the Aircraft Design Example

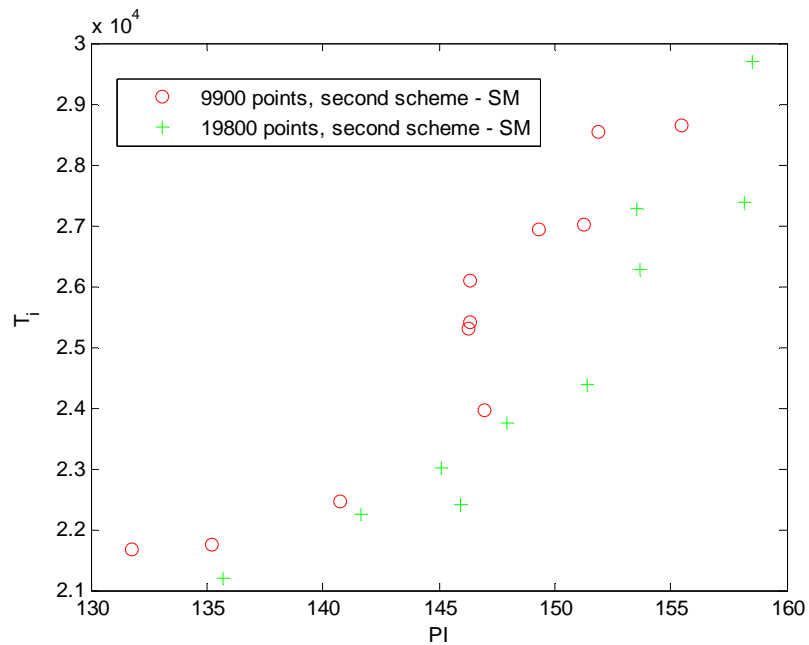


Figure 5-44: Comparison between the Two WPF's Found by the Second Search Schemes with 9900 and 19800 Points of S_2 and Surrogate Models for the Aircraft Design Example

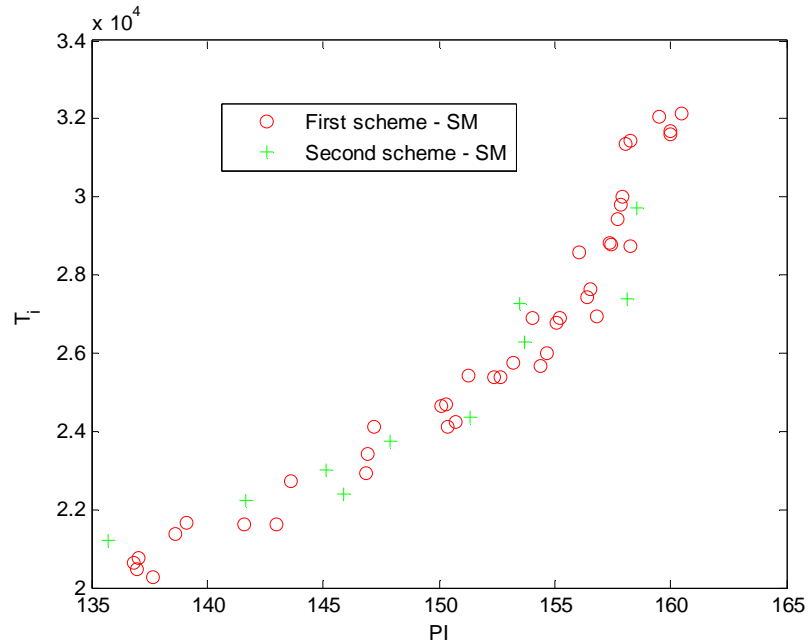


Figure 5-45: Comparison between the Two WPF's Found by the Two Search Schemes with 19800 Points of S_2 and Surrogate Models for the Aircraft Design Example

For the purpose of validation, the original CA's are also used to find the WPF. Both search schemes are executed with 19800 points of S_2 . Then two WPF's are obtained. These two new WPF's are compared with each other, and also compared with the corresponding one found with the surrogate models, respectively. These comparisons are given in Figure 5-46 – Figure 5-48.

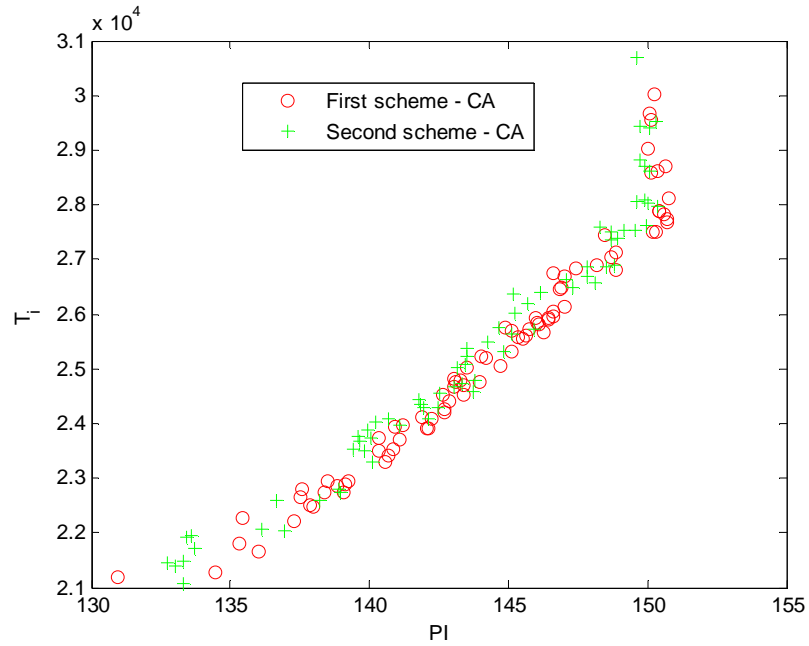


Figure 5-46: Comparison between the Two WPF's Found by the Two Search Schemes with 19800 Points of S_2 and Original CA's for the Aircraft Design Example

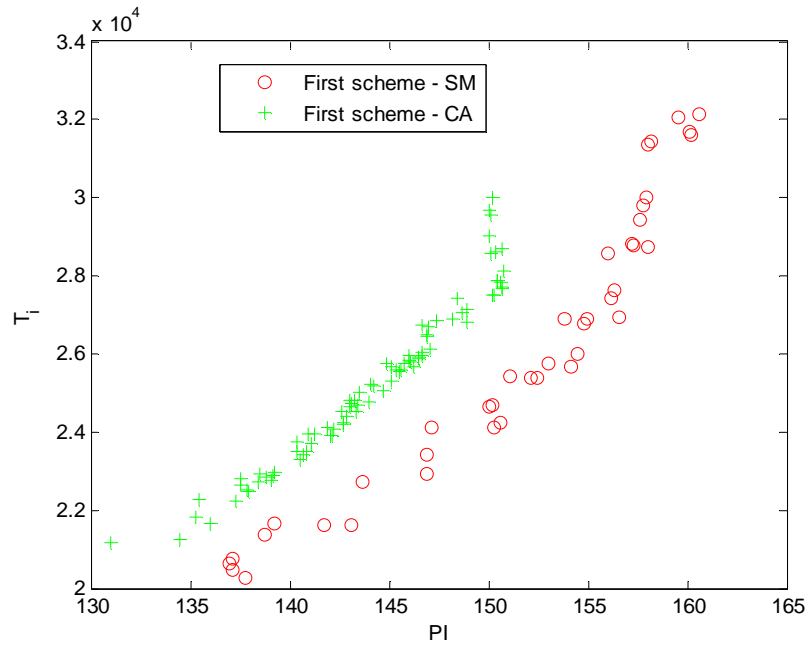


Figure 5-47: Comparison between the Two WPF's Found by the First Search Schemes with 19800 Points of S_2 , Surrogate Models, and Original CA's for the Aircraft Design Example

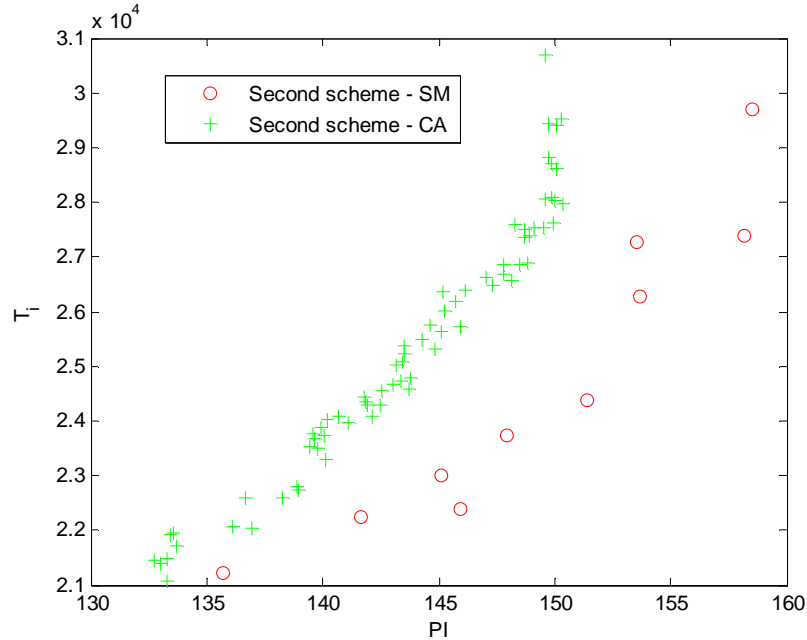


Figure 5-48: Comparison between the Two WPF's Found by the Second Search Schemes with 19800 Points of S_2 , Surrogate Models, and Original CA's for the Aircraft Design Example

5.3.3 Discussion

Observing and analyzing the design results of the aircraft design example, the following conclusions can be drawn:

1. It is important to choose an appropriate size of the search starting points S_2 . With the small size s_2 of 9900, the deterministic optimization problem with T_i as the single objective can not find a solution close to the known exact result (compare Table 9 with Case 1 in Table 14 and Table 15). With the first search scheme, the values of T_i are very close to that of the exact solutions, but the values of the design variables S and W_{to} are far away from those of the exact solutions. With the second search scheme, the values of T_i are a little far from that of the exact solutions. Additionally, the WPF's found with the small size s_2 of 9900 are not the correct ones, referring to Figure 5-43 and Figure 5-44.

2. The estimation of the size of S_2 given by Equation 4.13 is adequate for a good result. With the size s_2 of 19800 given by Equation 4.13, solutions very close to the exact solutions are found for both single-objective optimal problems, comparing the results of Case 2 with the exact solutions (compare Table 9 with Case 2 in Table 14 and Table 15); the WPF's found by two search schemes are very similar, referring to Figure 5-45; and the WPF points are uniformly distributed and the number of WPF points is enough for practical use. These justify the assumption made in section 4.2.4 that if the size of S_2 is large enough, the WPF or near WPF can be found.

3. Relaxation of the constraints and the convergence criteria is necessary. From other experiments (not recorded in this thesis) for this aircraft example, it has been found that no solutions satisfying all the constraints and convergence criteria can be obtained with the surrogate models and zero tolerance, whereas plenty solutions have been obtained with the original CA's and zero tolerance.

One note is that the relaxation tolerance is small with respect to the magnitude of the response, usually less than 1%. For example, $RMSE_{RCV}$ of V_{br} is 0.30913, then the relaxation tolerance is 0.61826 for this coupling variable; considering the magnitude of V_{br} is more than 500, one can this relaxation is very small.

4. With a small sample size, i.e. 150, the hybrid method can achieve accurate result for many responses. The MFE's of the hybrid models are less than 0.2%, and the MPE's less than 2% (see the results of hybrid models in Table 10 – Table 13).

5. The model selection advisor works very well. The model selection advisor selects different methods for different responses, all the MFE's are less than 1%, the maximum MPE less than 3%, and most MPE's less than 1% (see Table 10 – Table 13).

6. The Random Cross Validation method can provide good estimation for model predicting error. Comparing the estimation $\overline{\text{RMSE}}_{\text{RCV,Hybrid}}$ with true error $\overline{\text{RMSE}}_{\text{MPE}}$ in Table 10 – Table 13, one can see the values are close, especially for the models of SVR and the hybrid method.

7. The best objective values of PI and T_i found by the two search scheme with the surrogate models are better than those of the exact solutions, comparing Table 9 with Table 14 and Table 15, and referring to Figure 5-47 and Figure 5-48. The maximum difference for PI and T_i is about 7%.

The reason is that the errors of the surrogate models and the relaxation of the constraints and convergence criteria for the coupling variables lead to solutions that are inferior to the exact solutions in terms of satisfying the constraints and the convergence criteria. However, the errors of the surrogate model can not be eliminated; and as mentioned previously, the relaxation has to be made, otherwise too many or even all true valid design solutions may be excluded. From other experiments (not recorded in this thesis) with smaller and zero tolerance for this aircraft design example, it has been found that the errors of the surrogate models are the main reason for this difference, since smaller and zero tolerances do reduce this difference, but not always.

8. It can be concluded that the correct WPF is found with either search scheme for three reasons. The first is that the WPF's found by the two different search schemes with either surrogate models or original CA's are very similar. The second is that the WPF's found with the surrogate models are close to and have similar tendency as those with the original CA's, referring to Figure 5-45 – Figure 5-48. The last is that the best values of PI found with both schemes and the original CA's are no more than 151.5 and almost

vertical lines are found in Figure 5-47 and Figure 5-48. This phenomenon is the same as the results obtained with a conventional optimizer and original CA's with many different search starting points. Thus this aircraft design example shows the feasibility of this new framework.

5.4 A Reusable Launch Vehicle Design Example

Another simple yet typical reusable launch vehicle (RLV) design problem is used to show the feasibility of the new framework of determination of the WPF solutions under probabilistic constraints. This design problem is a typical multimodal problem; there are many local extremes for each objective. With this RLV design problem, the framework is shown to be able to handle multimodal problems. This problem has

- 1) 5 disciplinary analyses;
- 2) 9 system level design variables including 4 coupling variables that are assumed to be normally distributed about the mean values with 0.5σ symmetrical truncation (see APPENDIX G for a summary of the doubly-truncated normal distribution);
- 3) 1 PC's with required POS of 0.30, and
- 4) 2 design objectives. See APPENDIX I for detailed information.

The reason that the required POS is so small is because this design problem is quite sensitive to the perturbation around the converged design solutions, i.e. a small perturbation to a converged design solution will easily result in a non-converged design combination. Although there is only one probabilistic constraint, this RLV design problem still can demonstrate the feasibility of the proposed framework since the number of probabilistic constraints makes no difference to the operation of the probability counting Equation 2.55. Figure 5-49 shows the DSM of this problem.

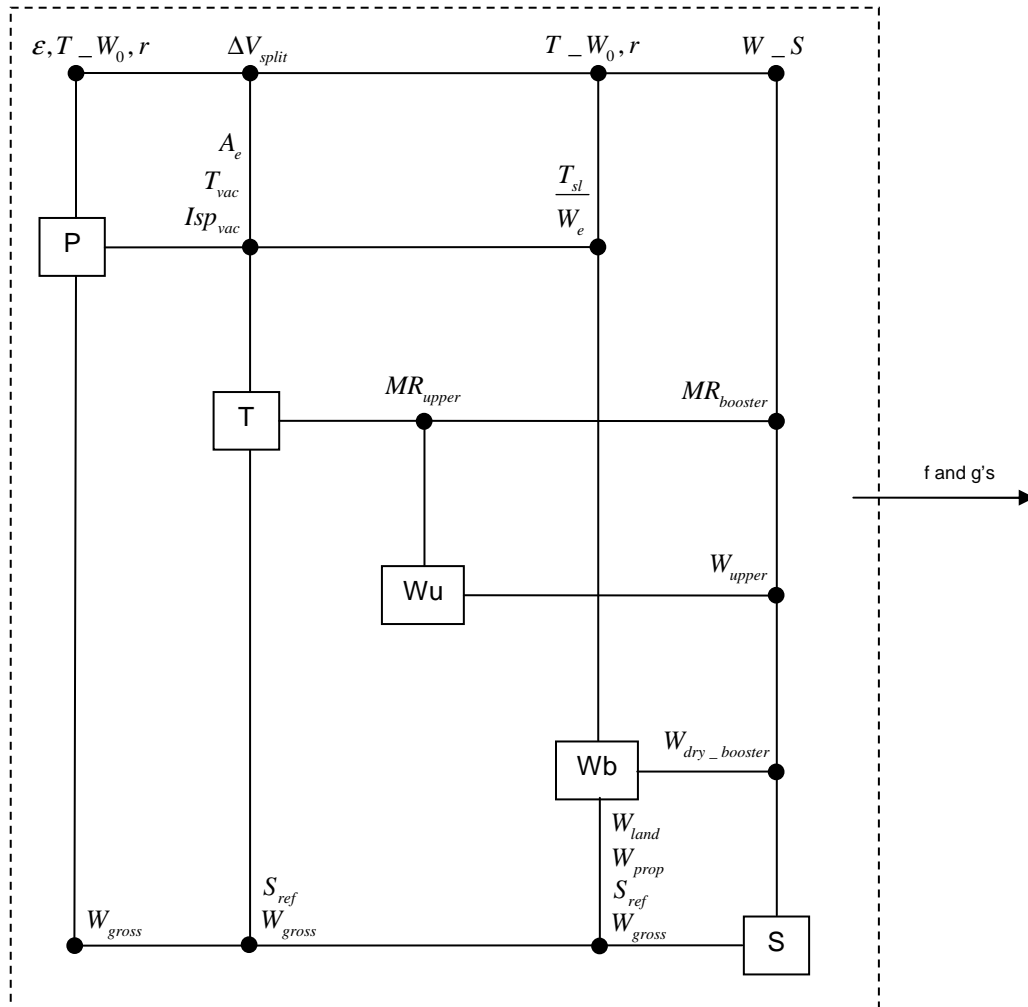


Figure 5-49: Example of the DSM of a Multidisciplinary RLV Design Problem

The surrogate models are first constructed, and then those models are used to find the WPF and its design solutions. For the purpose of validation, the original CA's are also used to find the WPF, the two kinds of WPF's obtained with the surrogate models and original CA's are compared. One set of exact single-objective deterministic (local) optimal solutions and objective values over the given design space are obtained and given in Table 16. Here 'exact' implicates the original CA's are used. Since this RLV design problem is a multimodal problem, the solutions in this table should be used with the search starting point with which the solutions are found. With the OBD method, the

search starting point used is $\varepsilon = 40$, $T_W_0 = 1.3$, $r = 6$, $W_S = 40$, $\Delta V_{split} = 0.5$,
 $S_{ref} = 4,500$, $W_{gross} = 2,000,000$, $W_{prop} = 1,750,000$, and $W_{land} = 200,000$.

Table 16: Exact Single-Objective Deterministic Optimal Results of the RLV Design Example

Objective	Objective Value	ε	T_W_0	r	W_S	ΔV_{split}
W_{gross}	651693.6	55.1	1.49	7.06	50.0	0.40
Isp_{vac}	431.7	10.0	1.20	5.32	25.54	0.74

5.4.1 Surrogate Models

For this example, discipline-level surrogate models are constructed. Except for the Wu CA that has only one simple response, a surrogate model is constructed for each of the responses of the discipline analyses. For each response, the training sample for surrogate model construction includes 150 points by HS sampling, the sample for estimation of true MPE includes 300 points by LHC sampling, and the sample for RCV includes 200 points by LHC sampling. The values of both the design variables and the response are normalized, the kernel is GRBF, the parameters C and ε are estimated by the practical estimation method, the parameter σ is selected by minimizing the modified information criterion BICC, and the best surrogate model is selected by minimizing the modified information criterion BICC as well.

The selected methods and goodness of fit for the responses are listed in Table 17 – Table 20, where \overline{RMSE}_{Tm} is the normalized model fitting error, $\overline{RMSE}_{RCV,Hybrid}$ is the normalized estimation of model predicting error using Random Cross Validation, \overline{RMSE}_{MPE} is the normalized true model predicting error calculated with random samples,

and $RMSE_{RCV}$ is the (real) estimated model predicting error after de-normalization. The normalized values are actually percentage values since all responses are normalized to [0, 100]. The accuracy of the selected surrogate models is satisfactory, since the maximum normalized model predicting error is less than 5%, and most of errors are less than 1%. The results once again show that $\overline{RMSE}_{RCV,Hybrid}$ can provide reasonable estimation for \overline{RMSE}_{MPE} , i.e. the RCV method can provide reasonable estimation for the model predicting error.

Table 17: Values of General Parameters and Goodness of Fit for the P CA

Response	Method	\overline{RMSE}_{Tm}	\overline{RMSE}_{MPE}	$\overline{RMSE}_{RCV,Hybrid}$	$RMSE_{RCV}$
A_e	Hybrid	0.0506	0.0748	0.0718	2.794
T_{vac}	RSM	0.1667	0.1737	0.0000	591.290
Isp_{vac}	RSM	0.0000	0.0000	0.0000	0.000
$T_{sl} - W_e$	RSM	0.0000	0.0000	0.0000	0.000
p_e	RSM	0.0000	0.0000	0.0000	0.000

Table 18: Values of General Parameters and Goodness of Fit for the T CA

Response	Method	\overline{RMSE}_{Tm}	\overline{RMSE}_{MPE}	$\overline{RMSE}_{RCV,Hybrid}$	$RMSE_{RCV}$
$MR_{booster}$	RSM	2.6124	4.9476	0.0000	0.1640
MR_{upper}	Hybrid	0.0208	0.2749	0.2526	0.0061

Table 19: Values of General Parameters and Goodness of Fit for the Wb CA

Response	Method	$\overline{\text{RMSE}}_{\text{Tm}}$	$\overline{\text{RMSE}}_{\text{MPE}}$	$\overline{\text{RMSE}}_{\text{RCV,Hybrid}}$	RMSE_{RCV}
$W_{\text{dry_booster}}$	Hybrid	0.0190	0.3816	0.2679	470.933

Table 20: Values of General Parameters and Goodness of Fit for the S CA

Response	Method	$\overline{\text{RMSE}}_{\text{Tm}}$	$\overline{\text{RMSE}}_{\text{MPE}}$	$\overline{\text{RMSE}}_{\text{RCV,Hybrid}}$	RMSE_{RCV}
W_{gross}	RSM	0.0126	0.0233	0.0258	818.953
S_{ref}	Hybrid	0.0762	1.2993	1.1994	118.439
W_{prop}	RSM	0.0129	0.0264	0.0277	795.409
W_{land}	RSM	0.0033	0.0035	0.0033	6.271

5.4.2 Design Results

For this design example, the new framework is implemented with the two neighborhood search schemes. For each search scheme, the sample sizes of S_2 is 39600. The number 39600 is the estimation given by Equation 4.13 with 99% probability and 2% error. For each valid solution of S_3 , the sample size of S_4 is given as 172 estimated by Equation 2.44 with 30% probability and 1% error.

With the first search scheme, 10040 valid solutions S_3 are obtained. Then 6628 candidate points of S_5 are obtained. Finally, 46 WPF points are obtained. The figure of the WPF's in the objective space is shown in Figure 5-50.

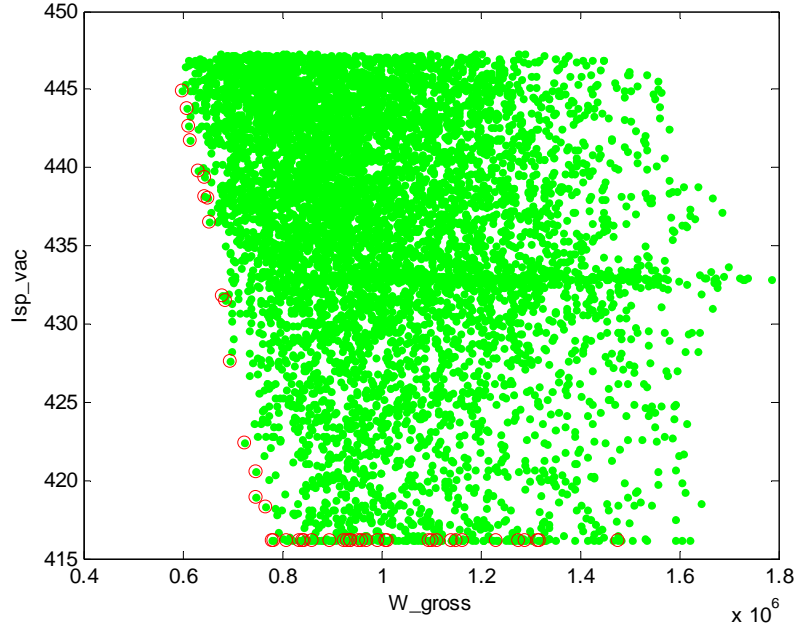


Figure 5-50: WPF Found by the First Search Scheme with 39600 points of S_2 and Surrogate Models for the RLV Design Example

With the first search scheme and surrogate models, the following valid solutions that are the closest to the single-objective deterministic optimal solutions from the given starting point are found and listed in Table 21. The distance used here to select the closest solution is the relative Euclidean distance.

Table 21: Valid Solutions Closest to Single-Objective Deterministic Optimal Solutions with the First Search Scheme and Surrogate Models for the RLV Design Example

Objective	Objective Value	ε	$T_{-}W_0$	r	$W_{-}S$	ΔV_{split}
W_{gross}	610,276.78* 649,855.84 ⁺	51.88	1.50	6.92	47.62	0.39
Isp_{vac}	431.23*, 431.23 ⁺	10.00	1.20	5.48	27.48	0.72

Note: *Predicted by the surrogate models;

⁺Predicted by the original CA's.

With the second search scheme and surrogate models, 14891 valid solutions S_3 are obtained. Then 9698 candidate points of S_5 are obtained. Finally, 67 WPF points are obtained. The figure of the WPF's is shown in Figure 5-51.

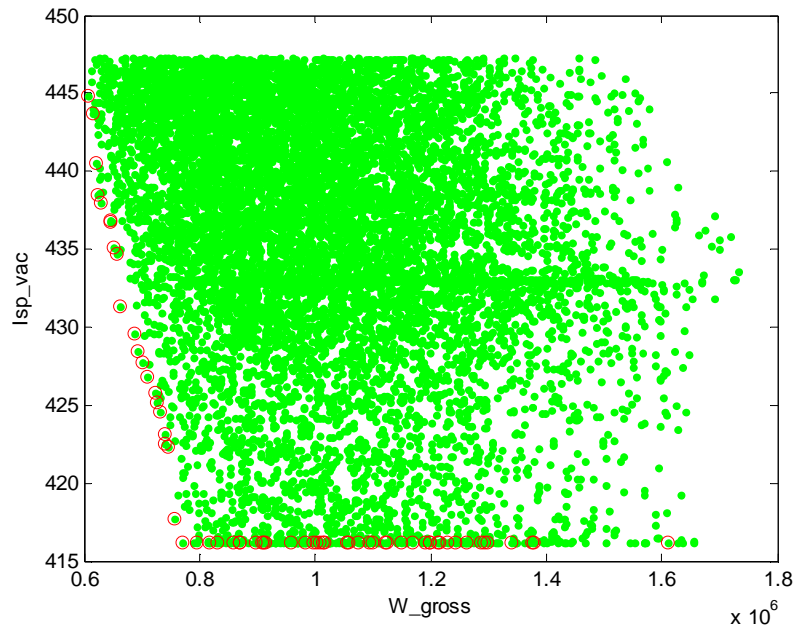


Figure 5-51: WPF Found by the Second Search Scheme with 39600 points of S_2 and Surrogate Models for the RLV Design Example

With the second search scheme, the following valid solutions that are the closest to the single-objective deterministic optimal solutions are found and listed in Table 22. The distance used here to select the closest solution is the relative Euclidean distance.

Table 22: Valid Solutions Closest to Single-Objective Deterministic Optimal Solutions with the Second Search Scheme and Surrogate Models for the RLV Design Example

Objective	Objective Value	ϵ	$T_{-}W_0$	r	$W_{-}S$	ΔV_{split}
W_{gross}	620,753.09* 651,971.82 ⁺	57.44	1.43	7.49	45.75	0.41
Isp_{vac}	431.07*, 431.07 ⁺	10.00	1.20	5.53	27.23	0.68

Note: *Predicted by the surrogate models;

+Predicted by the original CA's.

The two WPF's found by the two search schemes with 39600 points of S_2 and surrogate models are compared in Figure 5-52.

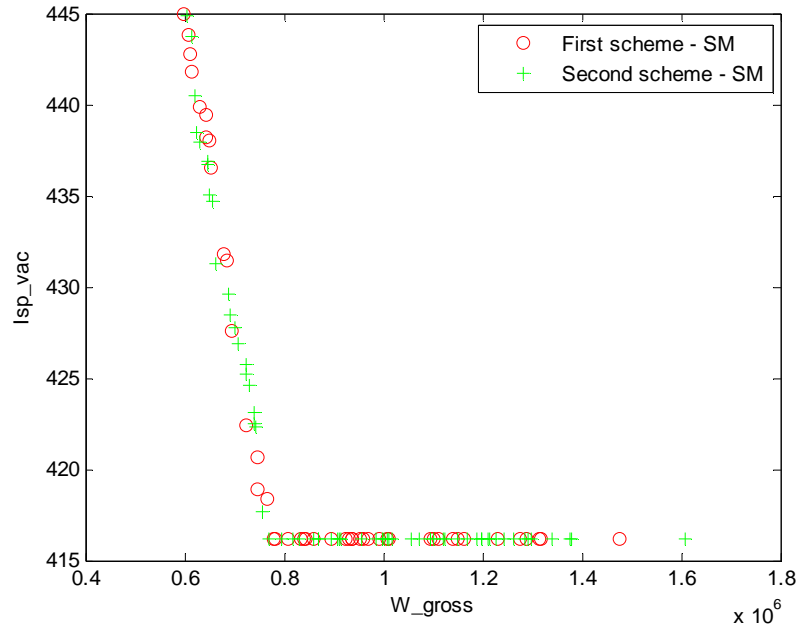


Figure 5-52: Comparison between the Two WPF's Found by the Two Search Schemes with 39600 Points of S_2 and Surrogate Models for the RLV Design Example

For the purpose of validation, the original CA's are also used to find the WPF. Both search schemes are executed with 39600 points of S_2 . Then two WPF's are obtained. These two new WPF's are compared with each other, and also compared with the corresponding one found with the surrogate models, respectively. These comparisons are given in Figure 5-53 – Figure 5-55.

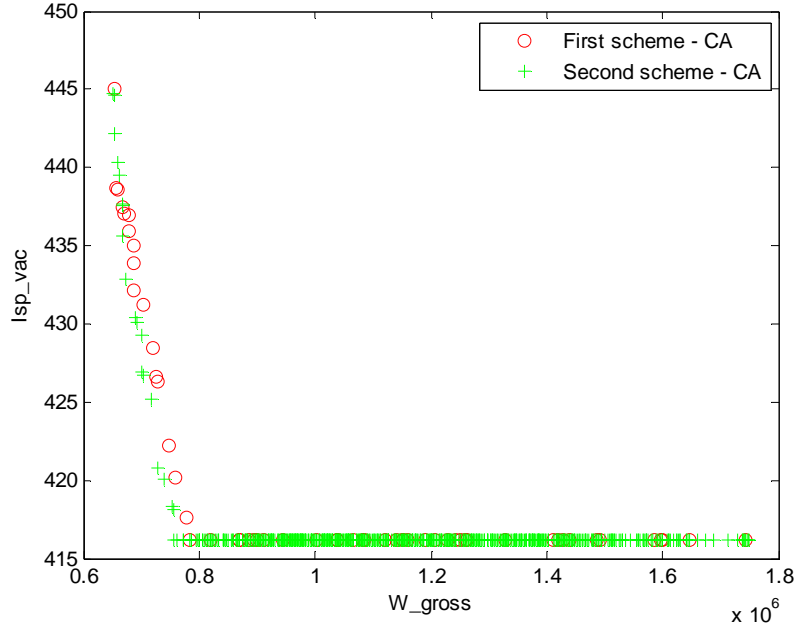


Figure 5-53: Comparison between the Two WPF's Found by the Two Search Schemes with 39600 Points of S_2 and Original CA's for the RLV Design Example

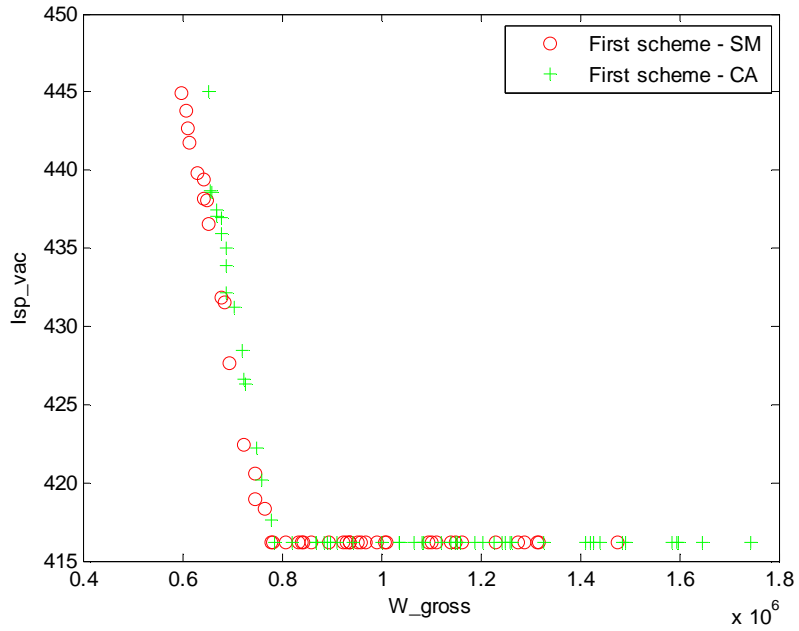


Figure 5-54: Comparison between the Two WPF's Found by the First Search Schemes with 39600 Points of S_2 , Surrogate Models, and Original CA's for the RLV Design Example

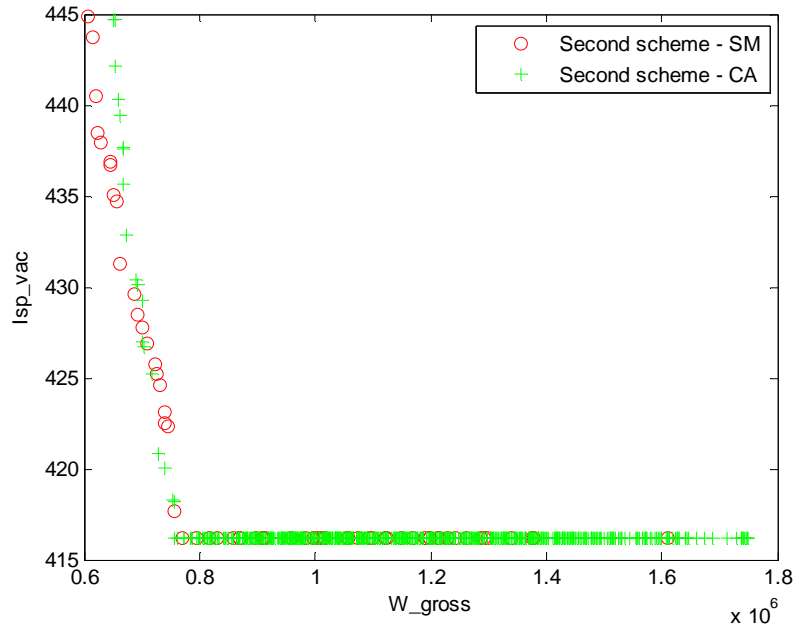


Figure 5-55: Comparison between the Two WPF’s Found by the Second Search Schemes with 39600 Points of S_2 , Surrogate Models, and Original CA’s for the RLV Design Example

5.4.3 Discussion

Observing and analyzing the design results of the reusable launch vehicle design example, the following conclusions can be drawn:

1. The estimation of the size of S_2 given by Equation 4.13 is adequate for good results. With the size s_2 of 39600 given by Equation 4.13, solutions very close to the exact solutions are found for both single-objective optimal problems, comparing the results with the exact solutions; the WPF’s found by two search schemes are very similar, referring to Figure 5-52; and the WPF points are uniformly distributed and the number of WPF points is enough for practical use. These again justify the assumption made in section 4.2.4 that if the size of S_2 is large enough, the WPF or near WPF can be found.

2. The WPF’s found with both search schemes and surrogate models are almost the same as those found with both search schemes and original CA’s, respectively, referring

to Figure 5-54 and Figure 5-55. The reason is that the errors of many surrogate models are very small, for example, the error of Isp_{vac} is zero.

3. Relaxation of the constraints and the convergence criteria is necessary. From other experiments (not recorded in this thesis) for this RLV example, it has been found that much less solutions satisfying all the constraints and convergence criteria can be obtained with the surrogate models and zero tolerance, whereas plenty solutions have been obtained with the original CA's and zero tolerance.

One note is that the relaxation tolerance is small with respect to the magnitude of the response, usually less than 1%. For example, $RMSE_{RCV}$ of W_{gross} is 819, then the relaxation tolerance is 1,638 for this coupling variable; considering the magnitude of W_{gross} is more than 200,000, one can this relaxation is very small.

4. With a small sample size, i.e. 150, the hybrid method can achieve accurate result for many responses. The MFE's of the hybrid models are less than 0.1%, and the MPE's less than 2% (see the results of hybrid models in Table 17 – Table 20).

5. The model selection advisor works very well. The model selection advisor selects different methods for different responses, all the MFE's are less than 1%, the maximum MPE less than 3%, and most MPE's less than 1% (see Table 17 – Table 20).

The results of Isp_{vac} , $T_{sl} - W_e$, and P_e can best show this. From APPENDIX I, the responses Isp_{vac} , $T_{sl} - W_e$, and P_e are explicitly constructed by RSM. The selected surrogate modeling methods for these three responses are just RSM and the errors of the surrogate models are zero, as shown in Table 17. The success of model selection also means the success of the modified information criteria and the random cross validation method for model predicting error estimation.

6. The Random Cross Validation method can provide good estimation for model predicting error. Comparing the estimation $\overline{\text{RMSE}}_{\text{RCV,Hybrid}}$ with true error $\overline{\text{RMSE}}_{\text{MPE}}$ in Table 17 – Table 20, one can see the values are close, especially for the models of SVR and the hybrid method.

7. The WPF's found with the second scheme have more points and the points are more evenly distributed than those found with the first scheme for this RLV example. This phenomenon shows that for different problems these two schemes may have different performance. Since one can not know if there is such a difference, it is better to use both schemes to solve the same problem and thus also obtain more solutions.

8. It can be concluded that the correct WPF is found with either search scheme, since the WPF's found by the two different search schemes with either surrogate models or original CA's are very similar, and the WPF's found with the surrogate models are very similar as those with the original CA's, referring to Figure 5-52 – Figure 5-55. Thus this RLV design example again shows the feasibility of this new framework.

9. The WPF figure can provide additional useful information to guide the design process besides helping the user choose design alternatives according to his/her preferences. For example, the WPF figures show that increasing $I_{sp_{vac}}$ will not help much reduce W_{gross} , and there is a lower limit for $I_{sp_{vac}}$ in order to satisfy the constraint. This kind of information can help the designer choose the right scope and direction to explore the design space, and thus reduce design time.

6 CONCLUSIONS AND RECOMMENDATIONS

The realistic conceptual design problem of complex systems is characterized by multiple disciplines, multiple objectives, uncertainties, and a short period for decision making. Because of the complexity of this design problem and the limitations of solving techniques available, this design problem traditionally was simplified to a problem of a combination of only some of the first three features, and also by simplification the design could be finished in a short period of time. To address this deficiency, a novel systematic framework has been formulated to consider all the first three features of a realistic conceptual design problem and solve this problem in a short period of time. This framework has been successfully implemented for a transportation airplane design problem and a reusable launch vehicle design problem. Besides, lower level problems have been solved in order to demonstrate the advantage of or validate some new techniques developed for this new framework, such as the hybrid surrogate modeling of RSM and SVR, the model selection advisor, and the new neighborhood search method.

In this section, the implementation results of design of complex systems and other exercises are surveyed to answer the driving research questions, emphasize the significance of the new framework, and make recommendations for future work and applications.

6.1 Research Questions

The research questions posed in Chapter 3 are actually used to guide the development of the framework. Now those questions are revisited and answered, based on the results obtained in this research.

1. Although it has many advantages and good characteristics, can SVR be used directly in engineering problems like RSM has been impressively demonstrated in the past? Or what means should be taken to make it suitable?

Answer: If the kernel function is the Gaussian radial basis function, it has been found by the author that SVR may have the numerical problem when the values of a design variable are large. This is because the GRBF is a negative exponential function. When values of a design variable are large or the exponent is small, the computer program may underflow. Except this, SVR can directly be used in engineering problems. One way to eliminate this limitation is to normalize the values of the design variables, as done in this research.

2. How can RSM and SVR be combined to form a new hybrid surrogate-modeling method that is accurate for many types of problems with a small training sample?

Answer: The way that the two methods are combined in this research is that first RSM is used to fit the model, and the RSM partial model is obtained; then the errors of the RSM partial model are fitted by the SVR, and the SVR partial model is obtained; last, the combination of the RSM partial model and the SVR partial model is the new hybrid model of RSM and SVR. The results in this research show this hybrid method is accurate for many different responses that are constructed with small training samples (sample size is no more than 150). The reason is that the RSM can capture the global tendency very well and the SVR can capture the local nonlinear behavior very well.

3. Using the previous five criteria for comparison in Chapter 2, is this hybrid method of RSM and SVR better than RSM or SVR for engineering problems? Or under what situation is it better?

Answer: The five criteria used to assess the methods' effectiveness are accuracy, efficiency, transparency, simplicity, and vulnerability to the problem of "curse of dimensionality". In terms of accuracy, the hybrid method can be better, especially when the responses have high nonlinear behaviors as shown in the Rastrigin example. For the other four criteria, the hybrid method is almost as good as RSM or SVR.

4. Is it possible to quantify the five criteria, such that the above comparison in Question 3 can be reliably made?

Answer: Accuracy, efficiency, and simplicity can be quantified. However, only accuracy and simplicity are quantified in this research and used for the comparison in Question 3 above.

5. Is it possible to create and formulate a process for which all pre-specified parameters of SVR can be determined automatically such that this hybrid surrogate-modeling method is simple to use as RSM?

Answer: A process has been created and formulated to **automatically** determine the three parameters of SVR, and the results are very good, as shown in this research. The process includes normalization of values of design variables and responses, practical selection of two parameters, and optimal selection of the third parameter by minimizing the modified information criterion.

6. Is there a kernel function for SVR that can work well for all engineering problems? If not, how to select a kernel function for different problems?

Answer: It has been found by other researchers and confirmed by the results in this research that the Gaussian radial basis function is such a kernel function for SVR.

However, as shown in this research, a normalization step is required for this kernel function to be applied to all engineering problems in order to avoid numerical difficulties.

7. What is the best data sampling technique for this hybrid surrogate-modeling method?

Answer: the Hammersley Sequence sampling technique is chosen as the sampling technique for the surrogate modeling methods in this research, including the hybrid method. The HS is chosen because the user can freely decide the number of sample points, the correlation is very low, and the sample points can be repeated for the purpose of comparison. The results in this research show high accuracy is obtained with the Hammersley Sequence sampling technique.

8. What quantitative measures of model accuracy and complexity are appropriate for the purpose of selection of surrogate-modeling techniques?

Answer: The model accuracy can be measured by the model fitting error and model predicting error, and the model complexity can be measured by the number of the parameters to be estimated. The model fitting error and model predicting error are quantitatively measured by the root mean square error in this research.

9. What is the proper way to combine the measures of model accuracy and complexity together so that a balance is achieved between these two kinds of measures?

Answer: In this research, the modified information criteria are used to combine the measures of model accuracy and complexity together and balance these two model measures. The results obtained in this research show that this way of combination is feasible and works very well.

10. When the accuracy is at the same level, can the selection criterion select the surrogate model constructed with a simpler surrogate-modeling method?

Answer: as shown in the RLV example, the selection criterion, such as the modified information criterion BICC, does select the simpler method, like RSM versus SVR or the hybrid, and SVR versus the hybrid. One can see that the hybrid models are very accurate when the hybrid method is selected for some responses.

11. At which level is the surrogate model constructed, i.e. at disciplinary or system level?

Answer: In this research, it is suggested that the surrogate models should be constructed at the disciplinary level in order that the relationships between the design variables and the responses have physical meanings.

12. How can a consistent design solution be found with this framework?

Answer: In this research, a new Monte Carlo simulation based neighborhood search method executed with optimizers is used to find consistent designs. Two search schemes are formulated.

13. Can the optimal consistent design solutions of the single-objective optimization problems with deterministic constraints be found, or near solutions be found?

Answer: The examples of the transport airplane design and the RLV design show that at least the near solutions can be found. Besides, the mathematical examples of finding WPF show that the exact solutions of the single-objective optimization problems can be found, although there are no coupling variables in these examples.

14. How can the WPF of each disjointed consistent design zone be found?

Answer: As long as the number of the search starting points is large enough, some WPF points of each disjointed consistent design can be found with the new Monte Carlo simulation based neighborhood search method. After sampling points in the objective space are obtained, the WPF points are picked out.

15. How can the number of search starting points be selected such that an appropriate number of WPF points can be found?

Answer: A rule-of-thumb equation is given to estimate the required number of sampling points for the new Monte Carlo simulation based neighborhood search method. This equation is based on an equation of Monte Carlo sampling for statistical inference. The design examples in this research show that the number of sampling points estimated with this equation is adequate to find appropriate number of WPF points.

16. How can evenly distributed WPF points be found for practical usefulness?

Answer: From a uniform sample of search starting points, the new neighborhood search method will lead to evenly or almost evenly distributed WPF points, as shown in the pure mathematical examples of finding WPF and the two complex system design examples.

17. Because of the errors introduced by the surrogate models, how can the thresholds in the PC's be relaxed such that trustable probabilities can be obtained?

Answer: The thresholds can be relaxed based on the model predicting errors of the surrogate models. A rule-of-thumb equation of the tolerance for relaxation is given in this research.

18. What is the best scheme for this new framework in terms of ability to find WPF solutions and computational time?

Answer: Two search schemes are formulated for the new Monte Carlo simulation based neighborhood search method. Both schemes can find the WPF and corresponding solutions, but usually the first scheme is faster than the second one, because the optimization problems of the first one are simpler than those of the second one. Therefore, in terms of computational time, the first one is better. However, both schemes should be used since these two schemes will find different WPF solutions; and on the other hand, theoretically the second scheme will find a probabilistic WPF that is closer to the true probabilistic WPF, since it directly searches the deterministic WPF corresponding to a neighborhood.

6.2 Summary of Contributions

The main contribution of this research is the development of a suitable framework to determine WPF solutions under probabilistic constraints for realistic conceptual designs of complex systems. Additionally, several new capabilities are created in order to formulate the framework. Those contributions are now summarized.

1. A systematic framework for determination of WPF solutions under probabilistic constraints for realistic conceptual design of complex systems. This framework is very unique as a whole because it enables solving a realistic conceptual design problem of a complex system in the context of multiple disciplines with coupling variables, multiple conflicting objectives, and uncertainties. This capability, to the best knowledge of the author, is the first of the kind.

2. A new Monte Carlo simulation based neighborhood search method. The new way of defining the neighborhoods around the search starting points generated by the Monte Carlo sampling method is one key to the success of the neighborhood search method and

the framework. Based on this, the new neighborhood search method can handle the situation of spatially disjointed consistent design zones, and decouple the process to find WPF in the objective space and corresponding solutions that satisfy the probabilistic constraints. Besides, it can be executed in parallel on different computers. In other words, this method is a new approach of integration and coordination of complex system design.

3. Automation of SVR. SVR is a state-of-the-art surrogate modeling method. It has very good performance for many types of problems. It has not been widely used in the aerospace industry because there are three parameters to be pre-selected and traditionally only the experts of SVR are capable of doing this work. Now all the three parameters can be automatically determined using the method developed in this research based on the information extracted from the sample. Since many design methods rely on surrogate models and SVR can provide accurate models in general, the automation of SVR enables more and more non-experts to use this advanced surrogate modeling method to improve the design results.

4. A new hybrid surrogate modeling method of RSM and SVR. Although SVR is very good, it has been found sometimes it is still not accurate enough for some engineering problems. By combining RSM and SVR together with RSM capturing the global tendency and SVR capturing the local high nonlinear behavior, the new hybrid surrogate modeling method of RSM and SVR can further improve the accuracy for problems of which SVR individually can not obtain satisfactory results with a small training sample.

5. A new approach for model predicting error estimation. There are mainly two ways to estimate the model predicting error of a surrogate model, i.e. using new random

cases obtained with the complex physical model and using a re-sampling method such as cross-validation or bootstrapping. Since it is time-consuming to run the complex physical model, the second way of re-sampling method is preferred. However, the traditional re-sampling methods still need a non-trivial period of time to run, and thus it can not be used for the purposes of selection of one parameter of the SVR and surrogate model selection. A new method, called random cross validation, is developed in this research to estimate the model predicting error very quickly. Although there is no theoretical proof yet, the results show that this new method can give very good estimation of the model predicting error. In this research, this method is investigated and limited to the models of RSM, SVR, and the hybrid method of RSM and SVR, but it can be investigated with other surrogate modeling methods in the future to extend its usage.

6. A model selection advisor based on modified information criteria. For a given problem, if the accuracy obtained by different surrogate modeling methods is similar (not necessary to be the same), the simpler surrogate model should be selected. Since the accuracy of a surrogate model is measured by the model fitting and model predicting error and the simplicity of the model can be measured by the number of parameters to be estimated, two conventional information criteria are modified in this research to achieve a balance among the model fitting error, model predicting error, and model simplicity, such that the simpler surrogate model will be selected from the models with similar accuracy. Thus a model selection advisor is created in this research based on the new modified information criteria. The results in this research show that the model selection advisor works very well.

6.3 Recommendations

The framework for the first time enables solving a realistic conceptual design problem of complex systems. Strictly speaking, the framework can indeed solve a much more realistic problem than those which the traditional methods are solving, for example, surrogate models are used instead of the original CA's. Therefore, this framework still needs to be improved. Although the surrogate models have to be used in the future because the complexity computer models keeps pace with the development of the computer speed, several areas are identified where improvements and continued focus should be made in the future. These areas include: 1) adaptive sampling; 2) modeling temporal randomness; 3) multidimensional data visualization.

1. Adaptive sampling. In order to obtain a good representation of the WPF with enough and evenly distributed points, a large number of search starting points are used in this research and a rule-of-thumb equation is given to estimate this number. However, a phenomenon is noticed that the uniform distribution of search starting points in the design space may correspond to the nonuniform distribution of sampling points in the objective space. This phenomenon can result in that some parts of the WPF have crowded points while the other parts have much less points, as can be seen in the first pure mathematical example of finding WPF. It is possible that in some problems some parts of the WPF have too less points to be practically useful. Adaptive sampling is recommended to solve this problem. By identifying the clustering zones in the objective space, one can identify the less dense zones in the objective space; then go back to the corresponding zones in the design space and add more points in these zones. By doing this kind of adaptive sampling, not only the above problem of over-sparse parts of WPF can be

solved, but also less search starting points can be used so that the total computational time can be reduced. This is because a small number of additional search starting points in certain identified zones in the objective space will be needed instead of a much greater number of additional points otherwise scattering all over the whole design space.

2. Modeling temporal randomness. The framework allows a probabilistic distribution is assigned around a nominal value, thus it has the capability to model the spatial randomness. However, in a real conceptual design problem, some uncertainties may exhibit both spatial and temporal randomness. Neglecting such temporal behavior may result in overestimating the possibility of satisfying the probabilistic constraints and consequent design solutions that still can not accommodate to the uncertainties. In order to keep the total computational cost at a manageable level, one way to consider the temporal randomness may be scenario analysis. The first step can be selection of several critical scenarios; then find the WPF solutions for each of those scenarios; and finally pick out the solutions that satisfy all the requirements under any of those scenarios.

3. Multidimensional data visualization to aid the decision making process. The capability to find the WPF and corresponding solutions provides the decision maker the opportunity to make more educated decision. However, the much more information contained in the WPF and its solutions also makes the decision making process more difficult, i.e. facing more information, it is harder to make a tradeoff, needless to say a good tradeoff. Multi-attribute decision-making (MADM) techniques such as the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) [102] can be used to aid the decision making process to select the 'best' solution according to the preference order of the decision maker, but those methods are not good for

communication among the decision makers or between the decision makers and the designers since how to determine the preference order needs a big discussion. Besides, those MADM techniques are not convenient to help analyze the rich information contained in the WPF and its solutions to make a better decision and reduce the time and cost of the design process. Multidimensional data visualization (MDDV) techniques are better ways for this kind of communication and data analysis. One of the most popular methods is the scatter plot matrix, which is a set of two dimensional scatter plots projected from high dimensional data. However, most of the MDDV techniques have various limitations, such as some of those techniques are difficult to understand, or computationally expensive, or not intuitive [103]. Recently a new MDDV technique, named hyper-space diagonal counting (HSDC) [104], is developed to overcome the above limitations. It intuitively visualize high dimensional (more than three) data with a two or three dimensional figure without loss of the meaning of the data and the concept of neighborhood, for example, one can display the information of two objectives on one axis, the information of two design variables on the second axis, and the information of the other three design variables on the third axis. This HSDC is recommended to be introduced and used with the framework to aid the decision making process.

APPENDIX A: SAMPLING METHODS OVERVIEW AND SOME MODERN METHODS

A.1 Overview of Sampling Methods

One important step to construct a surrogate model is to obtain the sample data. Because the computer model is like a black box for us, the only way to obtain knowledge about the computer model is through the sample data. However, with limited time and computational resources one can not explore the entire design space of the computer model for this purpose. Therefore, which point to be chosen needs wise decisions. Sampling methods or designs of experiments are developed to wisely choose the sample points. Although the research on sampling methods dates back to the early 20th century and has made abundant achievements, there is still ongoing research in this field focusing on modern sampling methods, such as Ref. [20], [23], [28], [105], [106], [107], [108], and [109], to name a few.

An experimental design can be defined as “a test or series of tests in which purposeful changes are made to the input variables of a process or system so that we may observe and identify the reasons for change in the output responses” [110]. The methods to arrange or plan those “purposeful changes” are collectively known as sampling methods or Design of Experiments. The input variables to be changed during the experiment are also called design variables or factors, and often represented by a n dimensional vector. The n dimensional space defined by the lower and upper bounds of the n design variables is the design space, which often is only the region of interest. A design point is a specific instance of the n design variables within the design space, and is also called a point, or a sample point. Therefore, a sampling method or Design of

Experiment is in other words a procedure to choose a set of sample points in the design space.

A response is a measured or evaluated quantity of the system or process corresponding to a specific sample point. A sample pair is the combination of a sample point and its response, and a sample is the collection of the sample pairs. The sampling procedure is formulated such that maximum trend information is gained from a limited number of sample pairs. This trend information is about the relationship between a response and the vector of the design variables. A response surface is any function that represents the “true” relationship over the design space. Sometimes a “response surface” refers in particular to a low-order polynomial function. Strictly speaking, such a polynomial function should be called a response surface approximation, which means any user-defined function as an approximation to the usually unknown true relationship. A response surface approximation is often called a surrogate model (or metamodel).

A sampling method is different from another in the way or pattern the sample points are distributed over the design space. The sample point distribution pattern determines the number of sample points for an experimental design and largely affects the ability of an experimental design to reveal the true response surface and the accuracy of the consequent surrogate model. Montgomery has identified eleven criteria for a good experimental design [110]. The two most important criteria are identified in [23]. One is minimum design variable correlation. In fact, the correlation is a kind of measures of both uniformity and randomness of the distribution of the sampling points throughout the design space, and the more uniformly and randomly the sampling points are distributed, the better the space filling effect is. Another is that the sampling points should distribute

over the design space without “clustering” of sampling points or large regions of unexplored design space, as shown in Figure A-1, where the richness of sample space means the distribution pattern of the sampling points. In practice, the characteristics of a specific surrogate-modeling method should also be considered in order to select a proper sampling method such that a minimum number of sampling points are required, for example, a D-optimal design is often chosen for the RSM.

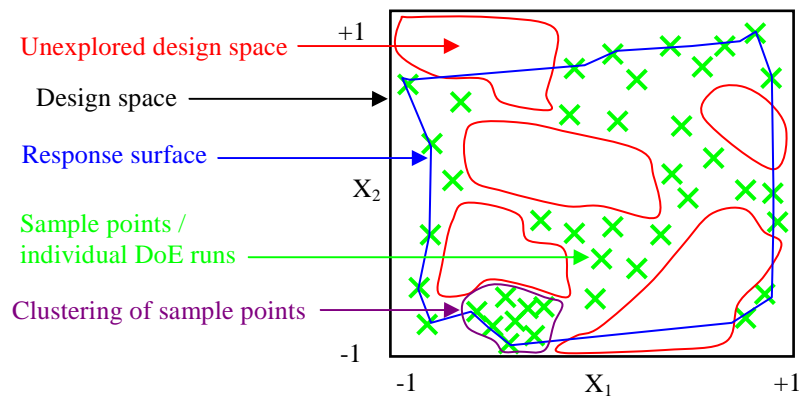


Figure A-1: Example of Richness of Sample Space [23]

The sampling methods nowadays can be divided into two groups, classical DoE's and modern DoE's. The classical DoE's were developed for laboratory and field experiments, such as biological and agricultural experiments. The modern DoE's were specifically developed for deterministic computer experiments or simulations. The main differences between the classical and modern DoE's are summarized in Ref. [20]. The fundamental difference is that classical DoE's assume there are random errors to be handled with, whereas modern DoE's assume no random errors. Therefore, classical DoE's generally put sample points at the extremes of the design space to minimize the effects of the random errors, whereas modern DoE's generally put sample points throughout the design space (space filling). Other techniques used by classical DoE's to

minimize the effects of random errors are experimental blocking, replication, and randomization, whereas modern DoE's do not need those techniques. Another main difference is that classical DoE's typically assume the possible values of a design variable are uniformly distributed between a lower and upper bound, whereas modern DoE's assume both uniform and non-uniform (such as Gaussian, Weibull, exponential) distributions. A common attribute between classic and modern DoE's is that the sample points are independently generated and can be evaluated concurrently using a parallel computing technique.

While most of the sampling methods generate the sample points all at once (one-stage sampling methods), one sampling strategy is under developing to add in new sampling points sequentially based on the information gathered from the earlier created surrogate model [21, 111]. This sampling strategy is called sequential sampling. There are two main advantages for sequential sampling methods. First, those methods can improve the accuracy of a surrogate model in a narrowed, interested, design region without the waste of sampling points outside this region. This is useful for surrogate-model-based optimization with searching strategies, because only part of the design space will be identified and explored during the searching process. If one-stage sampling methods for the entire design space are used, the sampling points outside the interested region are wasted. Second, the user can monitor the accuracy of the surrogate model and decide when to stop the sampling process, and thus reduce the possibility of generating more sampling points than necessary. One disadvantage of this strategy is that extra computational costs are needed to decide which new sampling point to be selected or evaluate the accuracy of the intermediate surrogate models. Besides, there is no guarantee

that a sequential sampling method can improve the accuracy of global surrogate models compared to one-stage methods [111], because the information from the surrogate models created previously can be misleading depending on the sample points and the surrogate-modeling methods. However, for early design stages global surrogate models are often used, thus one-stage sampling methods are used with maximum affordable sampling points for simplicity, instead of the sequential sampling methods.

A.2 Overview of Classical DoE's

The classical DoE's were first developed in the early 20th century for laboratory and field experiments, such as biological experiments or agricultural yield experiments. A common attribute among these experiments is that those experiments all have random error sources within the measured response. To minimize the effects of random errors, classical DoE's typically put sample points at or near the boundaries of the design space because by doing so, more reliable trend information can be extracted in the presence of random errors. A theoretical explanation for doing so is given in Ref. [41] and a simpler one in Ref. [20]. However, this leaves the interior of the design space largely unexplored.

The most often used classical DoE's are full and fractional factorial designs, central composite design, Box-Behnken design, and alphabetical optimal designs such as D-optimal design. The following Figure A-2 shows examples of some classical DoE's.

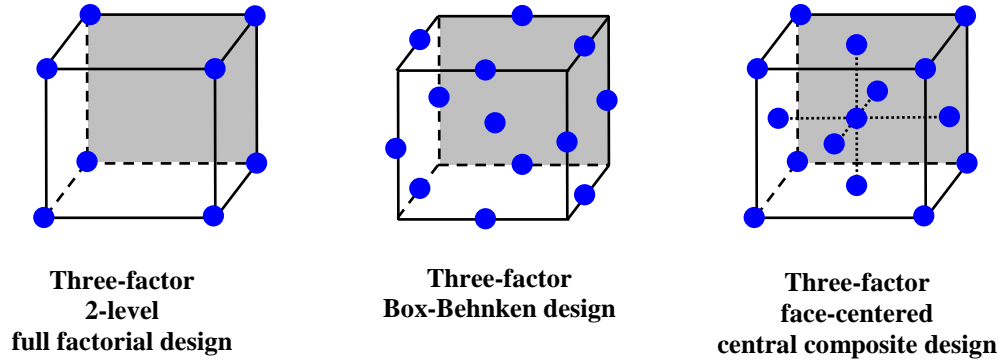


Figure A-2: Illustrations of Some Classical DoE's

Despite the originally intended use of those DoE's for laboratory and field experiments, classical DoE's can also be used for computer experiments – after removing the unnecessary blockings and replications. In fact, 2nd order RSM is often used with the classical central composite design or D-optimal design, because these designs are efficient for this surrogate-modeling method. “Efficient” here means there are few, if any, unnecessary sampling points.

A.3 Orthogonal Array Sampling

Orthogonal array sampling is a space filling sampling method that makes use of the orthogonal property of an orthogonal array to uniformly distribute the sampling points throughout the design space [112]. Its algorithm is as follows.

$$x_i^j = \frac{\pi_i(A_i^j) + U_i^j}{q}, \quad 1 \leq i \leq n, 1 \leq j \leq s \quad (\text{A. 1})$$

where x_i is the i^{th} design variable that is normalized to [0,1] from its original interval $[x_i^l, x_i^u]$, n is the number of design variables, s is the number of sample points, A is an orthogonal array (matrix), $\pi_i(A_i^j)$ denotes the $(i, j)^{\text{th}}$ element in A , U is a uniform random value on [0,1], and q is the number of bins or levels for each design

variable (q is the same for all design variables). The superscript j denotes the sample point number.

An orthogonal array (OA) A of strength t ($t < n$) is a matrix of s rows and n columns with every element being one of q numbers: $0, 1, \dots, q-1$, such that in any $s \times t$ sub-matrix each of the q^t possible rows occurs the same number λ of times, which is actually the definition of “orthogonal” here. λ is the index of the OA. Thus an OA is denoted by $OA(s, n, q, t)$ with $s = \lambda q^t$. The following matrix is an example of OA with $s = 4$, $n = 3$, $q = 2$, and $t = 2$ (thus $\lambda = 1$), and Figure A-3 shows the four sample points in a 3 dimensional design space generated by this $OA(4, 3, 2, 2)$.

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

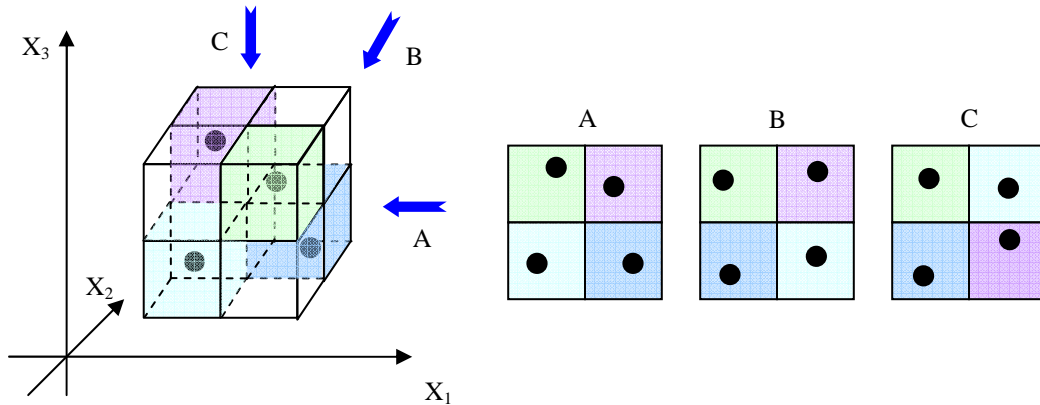


Figure A-3: Example of Three Dimensional Orthogonal Array Sampling

The OA sampling has two significant advantages. First, because of the orthogonal property of an OA, the sample points are uniformly distributed in any t dimensional projection of a n dimensional design space, see Figure A-3 for example. Second, unlike

LHC sampling and HS sampling, OA sampling may have sample points near the corners and/or boundaries of the design space. The OA sampling has two main disadvantages. First, the user can not freely decide the number of sample points if $t > 1$ (recall $s = \lambda q^t$). Second, the OA generation is not trivial and thus it is not convenient to generate an OA sample instantly. Usually frequently used OA's are published in tabular form and the user has to pick a proper OA table and input it into the computer for sample generation.

The Equation A.1 gives only the algorithm for design variables with uniform distributions, but similar to LHC sampling, the OA sampling can be used for design variables with non-uniform distributions.

A.4 Uniform Designs

Since the uniformity of the sample point distribution is important to the accuracy of the surrogate model, why does one not try to distribute the sample points uniformly throughout the design space directly? Uniform design is such a sampling method to scatter the sample points uniformly in the first place, and has been popularly used since 1980 [108, 113, 114].

The algorithm of UD begins with the measure of uniformity of the sample point distribution. Suppose C^n denotes the design space of the n design variables, $P_s = \{X_1, X_2, \dots, X_s\}$ denotes the set of the sample points, and $D_p(P_s)$ denotes the discrepancy of the empirical joint CDF $F_s(X)$ of P_s from a joint uniform CDF $F(X)$ on C^n . Then the uniform design of sample points is obtained by finding the ones that have the minimum discrepancy $D(P_s)$ over all possible s points on C^n .

The empirical joint CDF $F_s(X)$ is defined as:

$$F_s(X) = \frac{1}{s} \sum_{i=1}^s I\{X_i \leq X\} \quad (\text{A. 2})$$

where $I\{\cdot\}$ is the indicator function, giving 1 if the condition in the parenthesis is satisfied or 0 otherwise.

The most commonly used discrepancy function is defined as:

$$D(P_s) = \sup_{X \in C^n} |F_s(X) - F(X)| \quad (\text{A. 3})$$

Obviously it is not trivial to find the uniform design sample points given the sample size s and the design space C^n , therefore, like the OA's, the UD's are published in tabular form for use. However, for $n = 1$, the UD is easily given as

$$P_s = \left\{ \frac{1}{2s}, \frac{3}{2s}, \dots, \frac{2s-1}{2s} \right\} \quad (\text{A. 4})$$

with $D(P_s) = \frac{1}{2s}$. For $n > 1$, an approximate UD can be given, and is called a unique UD (UUD). The UUD can be shown to be the same as the lattice sampling (see Equation 2.7).

The UD sampling method has two significant advantages, i.e. uniformity of the sample point distribution and the freedom of the user to choose the number of sample points. It should be noted that the information of the distributions of the design variables are intentionally ignored in UD in order to get uniformly distributed sample points.

APPENDIX B: SURROGATE-MODELING PRELIMINARIES AND THREE MODERN METHODS

B.1 Statistical Inferences

Statistical inference can be defined as the science of deducing properties of an underlying (probability) distribution function of a random variable from a sample of a population, where the population is all possible observations available from this probability distribution and a sample is a particular subset of the population [115].

There are two main approaches to statistical inference:

1. The parametric or particular inference;
2. The non-parametric or general inference.

The parametric inference is the approach of statistical inference based upon a distributional assumption for the population, while the non-parametric inference does not make any such assumptions. Thus the non-parametric inference is also called “distribution-independent” inference.

The parametric inference is developed from the descriptive statistics that shows many events of reality can be well described by some simple distribution functions. For example, the dimensions of a product are in general normally distributed and the expected life of an electronic device is in general exponentially distributed. The creation of parametric inference is based on the following belief:

One knows the problem to be analyzed very well and can find some (specific) simple distribution functions or the combination of those distribution functions to describe the problem very well.

Then the task of parametric inference is to estimate the parameters of these assumed distribution functions, e.g. the mean and variance of a normal distribution, and/or the coefficients (parameters) in the combination function of those assumed functions, e.g. the coefficients in a polynomial function.

Although the parametric inference has been successfully used to solve many problems in different areas, its results depend on the validity of the distributional assumption on which it is based. If a normal distribution is made, whereas the unknown true distribution is skewed, then the inference result is misleading. This fact leads to the creation of non-parametric inference based on the following belief:

One does not have reliable a priori information about the distribution function underlying the problem to be analyzed or the problem is so different or complicated that it can not be described by only known simple distribution functions or combinations of those simple distributions, so it is necessary to find an approximation to the true one of the problem.

Then the task of non-parametric inference is to find a method for any problem that can infer an approximation function to the true distribution function from the given sample, without making assumptions about the distribution function.

One straightforward method of non-parametric inference is the empirical cumulative distribution function (CDF) method. The empirical CDF will converge to the true CDF with increasing sample size, and this is the result of the classical Law of Large Numbers. Another method is to establish a principle as a decision criterion to find a function or a combination of functions from a given set of functions (including the simple distributions and polynomials) that best approximates the unknown true distribution

function with increasing sample size. One such principle is the Empirical Risk Minimization (ERM) principle, which will be discussed later.

The non-parametric inference is developed because of the limitations of the parametric inference and is widely adopted in engineering and other sciences because of its general validity as a result of the weak distributional assumptions. However, it has to be indicated that the parametric inference should be used if the distributional assumption is valid for a real life problem, because it will provide a more precise or more powerful analysis than the corresponding non-parametric inference.

B.2 The Problem of “Curse of Dimensionality” of the Parametric Inference

The problem of “Curse of Dimensionality” is a shortcoming of parametric inference that was discovered in the 1960s when computers started to be widely used to analyze complex models that have a large number of design variables (factors) or obtain more accurate approximation. It was observed that the sample size and the computational resources are required to increase exponentially with the increase of the number of factors to be considered. This phenomenon is called by R. Bellman as the “curse of dimensionality” [44].

For example, the Weierstrass theorem states that any continuous function of n design variables can be approximated on a finite interval by polynomials with any degree of accuracy. However, this polynomial approximation can only guarantee the accuracy $O(N^{-s/n})$ [44], where N is the number of terms of the polynomial and s is the number of derivatives of the function to be approximated. Therefore, even if s is a small number, in order to obtain the desired level of accuracy the number of polynomial terms N has to be increased exponentially with the number of design variables n . Thus the sample size

and computer resources also have to increase exponentially in order to obtain the parameters in the polynomial approximation. Or in other words, the accuracy level of the parametric inference increases slowly with the increase of the number of polynomial terms and increase of the required sample size. Besides, if the number of polynomial terms is fixed, the increase of sample size can just lead to trivial or even no increase of the accuracy level.

Therefore, this problem of “curse of dimensionality” means that for the real life multivariate problems with dozens of or even hundreds of design variables, to obtain a good approximation one needs a large set of functions and a large required sample size, and one can not rely on increasing sample size to increase the accuracy level because the accuracy level increases slowly with increase of the sample size. This is a considerable limitation of the parametric inference and therefore many researchers now do not use it to do statistical inference.

B.3 Problem of Regression Estimation and Related Decision Principles

In this section the problem of regression estimation is described and the principles used to select the optimal regression function are introduced.

B.3.1 Problem of Regression Estimation

Regression is a method to obtain a mathematical relationship (function) between the mean or expected value of a response variable y and a vector X of predictor variables (x_1, x_2, \dots, x_n) based on a sample or a set of observed pairs [116]

$$S : \{(y_1, X_1), (y_2, X_2), \dots, (y_s, X_s)\}$$

where X_i is called a sample point, and (y_i, X_i) is called a sample pair. The values X_i of X are deterministic because the predictor variables are controllable, and values of

y have a random component and follow an unknown distribution because usually those values are observed results of real life phenomena [116].

The above sample is generated like this: let the vectors X_i 's appear randomly and independently according to a known or unknown fixed distribution $f_X(X)$, then for each X_i a value of y is selected according to a conditional distribution $f(y|X)$. In this case, there exists a joint distribution function $f(y, X) = f(y|X)f_X(X)$, which is unknown because at least the conditional distribution $f(y|X)$ is unknown.

This conditional distribution $f(y|X)$ actually describes the relationship between the response variable y and the predictor variable vector X . However, it is very difficult to estimate this conditional distribution based on the sample data. On the other hand, people are more interested in the expected or mean of y for the purpose of prediction:

$$r(X) = \int yf(y|X)dy \quad (\text{B. 1})$$

This function $r(X)$ is called the regression function, and the problem of its estimation based on the given sample data is called the problem of regression estimation.

Because of the unknown conditional distribution $f(y|X)$, it is still impossible to obtain the regression function $r(X)$. One can hope to obtain an approximation to the regression function by the following method [44]:

According to the characteristics of the problem studied, assume a family of functions $g(X, \theta)$, in which θ is called the parameter and is a scalar or a vector of scalars to be determined; Then under conditions

$$\iint y^2 f(y, X)dydX < \infty, \iint r^2(X)f(y, X)dydX < \infty$$

the problem of regression estimation is reduced to the problem of minimizing the following risk $R(\theta)$ based on the given sample data:

$$R(\theta) = \iint (y - g(X, \theta))^2 f(y, X) dy dX \quad (\text{B. 2})$$

If the (real) regression function $r(X)$ is in the function family $g(X, \theta)$, the minimum of risk $R(\theta)$ is attained at the regression function $r(X)$; If the regression function $r(X)$ is not in the family $g(X, \theta)$, the minimum of risk $R(\theta)$ is attained at the function $g(X, \theta^*)$ that is closest to the regression function $r(X)$ in the metric $L_2(P)$:

$$\rho(f_1(X), f_2(X)) = \sqrt{\int (f_1(X) - f_2(X))^2 f_X(X) dX}$$

To prove this, first denote:

$$\Delta g(X, \theta) = g(X, \theta) - r(X)$$

Then the Equation B.2 can be written as

$$\begin{aligned} R(\theta) &= \iint (y - r(X))^2 f(y, X) dy dX + \iint (\Delta g(X, \theta))^2 f(y, X) dy dX \\ &\quad - 2 \iint \Delta g(X, \theta) (y - r(X)) f(y, X) dy dX \end{aligned}$$

The third summand above is zero, because according to Equation B.1

$$\begin{aligned} &\iint \Delta g(X, \theta) (y - r(X)) f(y, X) dy dX \\ &= \int \Delta g(X, \theta) \left[\int y f(y|X) f_X(X) dy - \int r(X) f(y, X) dy \right] dX \\ &= \int \Delta g(X, \theta) \left[f_X(X) \int y f(y|X) dy - r(X) \int f(y, X) dy \right] dX \\ &= \int \Delta g(X, \theta) [f_X(X) r(X) - r(X) f_X(X)] dX = 0 \end{aligned}$$

Thus we get that

$$R(\theta) = \iint (y - r(X))^2 f(y, X) dy dX + \int (g(X, \theta) - r(X))^2 f_X(X) dX \quad (\text{B. 3})$$

Since the first summand is independent of parameter θ , the function $g(X, \theta^*)$ that minimizes the risk $R(\theta)$ is the regression function $r(\theta)$ if $r(\theta)$ is in the function family

$g(X, \theta)$; or it is the closest function to $r(\theta)$ in the function family in the metric $L_2(P)$ if $r(\theta)$ is not in the function family $g(X, \theta)$.

The loss function family is defined as

$$L(Z, \theta) = (y - g(X, \theta))^2 \quad (\text{B. 4})$$

where the vector Z consists of the variable y and the vector X , and thus has $(n + 1)$ elements.

The following Figure B-1 shows the simple univariate linear regression model, in which the sample points are shown as the black dots and all the conditional distributions $F(y|x)$ are assumed to follow the same $N(0, \sigma^2)$ distribution. In Figure B-1 the regression line passes the expected values or means of the normal distributions of the corresponding response values y_1, y_2, \dots, y_s , and the sample points scatter around this regression line.

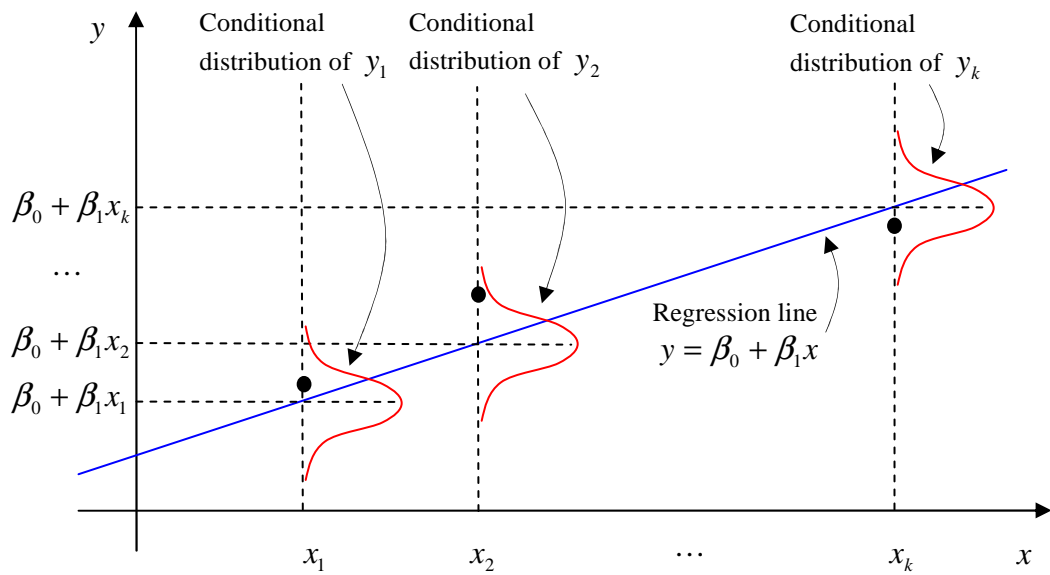


Figure B-1: Simple Univariate Linear Regression Model (Based on [115])

B.3.2 The Empirical Risk Minimization Principle

In section B.3.1, it has been shown that the regression problem can be reduced to the problem of selection of the best function in the assumed function family $g(X, \theta)$ by minimization of the risk $R(\theta)$. However, since the joint distribution $f(y, X)$ is unknown, the risk $R(\theta)$ in Equation B.2 still can not be minimized directly. Then, what other options are available? The answer is to minimize the empirical risk $R_{emp}(\theta)$.

Using the concept of loss function $L(Z, \theta)$ (note that the loss function is not limited to the form in Equation B.4, the Equation B.2 can be rewritten as

$$R(\theta) = \int L(Z, \theta) f(Z) dZ \quad (\text{B. 5})$$

Then the empirical risk $R_{emp}(\theta)$ is defined as

$$R_{emp}(\theta) = \frac{1}{s} \sum_{i=1}^s L(Z_i, \theta) \quad (\text{B. 6})$$

The minimization solution of the empirical risk $R_{emp}(\theta)$ is considered as an approximation to that of the true risk $R(\theta)$, and the principle of solving the empirical risk function as an approximation to the solution of the true risk function is called the Empirical Risk Minimization principle. Obviously, this principle is distribution independent. Therefore, it can be applied to many types of problems.

This ERM principle is possibly from the idea that the empirical CDF will converge to the true CDF with increasing sample size, although the empirical risk does not necessarily converge to the true risk with increasing sample size. However, it can be proved that some assumed function families $g(X, \theta)$ are necessary and sufficient for the empirical risk function to converge to the true risk function, and the rate of convergence depends on both the property of the assumed function and the sample size. Besides, this

principle does have its foundation: it was proposed based on summarization of the methods used in learning machines (computer programs) since the 1960's that can do more or less generalized regression (regression without knowing the distribution), and the success of these learning machines shows the effectiveness of this principle [44].

B.3.3 Model Selection and the Principle of “Occam’s Razor”

By minimizing the empirical risk function, the value of the parameter θ can be determined. This process is called parameter selection. A question is raised, how do we select the best function family $g(X, \theta)$? This question is raised because of the observations: a more complex model usually has more powerful representational capacity and can typically fit the sample data better, but is not necessary to provide better prediction for further/future data (the data outside of the sample data). This process of function family selection is called model selection.

A general philosophical principle known as “Occam’s razor” is used for model selection.

Entities should not be multiplied beyond necessity.

— “Occam’s razor” principle attributed to William Occam (c. 1285 – 1349).

The exact interpretation of Occam’s razor is under discussion. The most common one for model selection is: the unnecessarily complex models should not be preferred to simpler ones. It has to be pointed out that this interpretation does not always prefer simpler models; in fact it just does not like the “unnecessarily” complex ones, in other words, if the simpler model can provide **similar** level of accuracy, the complex one is not preferred; otherwise, if the accuracy of the simpler one is much worse than that of the

complex one, the complex one is preferred. Therefore, this principle needs to be used with other principles or methods that can balance accuracy and model complexity.

At the first look, one may think the principle of Occam's razor can not be proved mathematically, and thus its justification can only rely on two facts: first, people prefer a simpler model if the simpler one can fit the sample data as well as the complex one; second, this principle has been successfully applied to practice in the past. However, it can be shown that the Bayesian probability theory supports this principle quantitatively [117].

Suppose that two models M_1 and M_2 can fit a given sample data D to the same level of "goodness", and M_1 is simpler than M_2 . Now we want to know which model is the more probable one based on the sample data. This problem is equivalent to comparing two conditional probabilities: $P(M_1 | D)$ and $P(M_2 | D)$.

According to the Bayes' theorem, $P(M_1 | D) = P(M_1)P(D | M_1) / P(D)$, and $P(M_2 | D) = P(M_2)P(D | M_2) / P(D)$, where $P(M_i)$ is the prior probability of M_i and it reflects a person's subjective preference; $P(D | M_i)$ is the probability that the data set a model is based on happens to be the sample D if the model is M_i , and it is also called the evidence for M_i ; and $P(D)$ is the probability that the sample D is selected. This gives the following probability ratio:

$$\frac{P(M_1 | D)}{P(M_2 | D)} = \frac{P(M_1)P(D | M_1)}{P(M_2)P(D | M_2)} \quad (\text{B. 7})$$

As shown in Figure B-2, the complex model M_2 by its nature can fit a greater variety of data than the simpler one M_1 , therefore $P(D | M_1) > P(D | M_2)$. Suppose one

does not have preference for any of the two models, thus $P(M_1) = P(M_2)$. Therefore, according to Equation B.7, $P(M_1 | D) > P(M_2 | D)$, i.e. the simpler model M_1 is the more probable one.

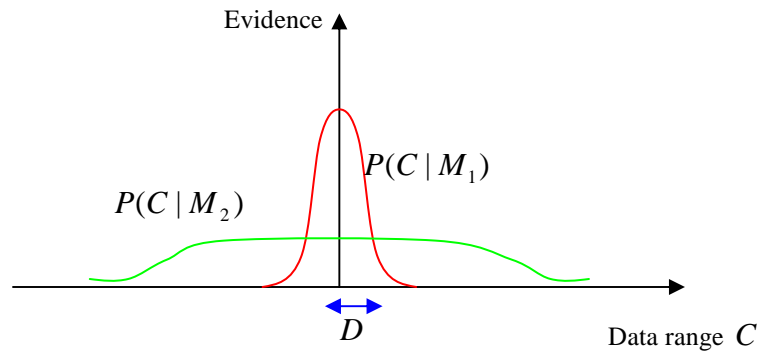


Figure B-2: Occam's Razor Is Supported by Bayes' Theorem [117]

One simple example to demonstrate Occam's razor is the selection between a model of $ax + b$ (a first order polynomial) and a model of $cx^2 + dx + e$ (a second order polynomial). Obviously, the second model can accurately fit not only samples generated by a first order polynomial but also samples generated by a second order polynomial, while the first model can accurately fit only samples generated by a first order polynomial. However, if both models can fit a given sample well, then the model of the first order polynomial is preferred.

B.3.4 The Structural Risk Minimization Principle

Although the ERM principle has been successfully applied to generalized regressions, it causes the problem of overfitting because of its implication to minimize the empirical risk function $R_{emp}(\theta)$ at any cost, e.g. the traditional ANN adopting this principle suffers this problem. Essentially the problem of overfitting is the consequence

of an unnecessarily complex model being selected. Therefore the principle of Occam's razor can be used to overcome this problem.

The combination of the ERM and Occam's razor principles led to a new principle, i.e. the Structural Risk Minimization (SRM) principle. In contrast to the ERM principle, this SRM principle minimizes an upper bound on the (empirical) risk function and thus finds the optimal compromise between the information amount of sample data, and the complexity (or accuracy, assuming the greater the complexity, the higher the accuracy) of the approximation of the sample data by the function chosen from the assumed function family $g(X, \theta)$. This compromise is achieved by capacity control, which is the embodiment of the Occam's razor principle. The "capacity" here can be considered as the capability of a function to make the empirical risk function $R_{emp}(\theta)$ converge to the true risk $R(\theta)$, and is not necessarily the number of parameters of the function family $g(X, \theta)$ [44].

The SRM principle has been first realized in the method of Support Vector Machine by V. Vapnik in late 1970's [45], and has been shown to be superior to the ERM principle. As the application of SVM to regression, the Support Vector Regression is gaining popularity due to many attractive features and promising empirical performance inherited from the SRM principle. SVR is one focus of this research work and will be introduced later.

B.4 Neural Network

The (artificial) neural network (ANN or simply NN) surrogate-modeling method is inspired by the way that biological neural networks (e.g. the brain) function that enables those networks to cognize and process new data from outside [61, 118]. To emulate the

functions of the biological neural networks, the physical structures are abstracted in a ANN as three kinds of layers: the input, the hidden, and the output layers, and each layer generally consists of a large number of simple processing units, which are called nodes corresponding to the neurons (Greek: nerve cells) in a biological neural network.

There is only one input layer and one output layer, but the number of hidden layers can be greater than one. Each node in one layer is connected to nodes in other layers in a specific way, and there is no interconnection between any two nodes in the same layer in a non-recurrent ANN, as in the example shown in Figure B-3. Therefore, the structure of a non-recurrent ANN is just a much simplified version of the real biological neural network. Although a recurrent ANN mimics a biological NN better with connections among all nodes, it is too complicated and not popular as a non-recurrent ANN.

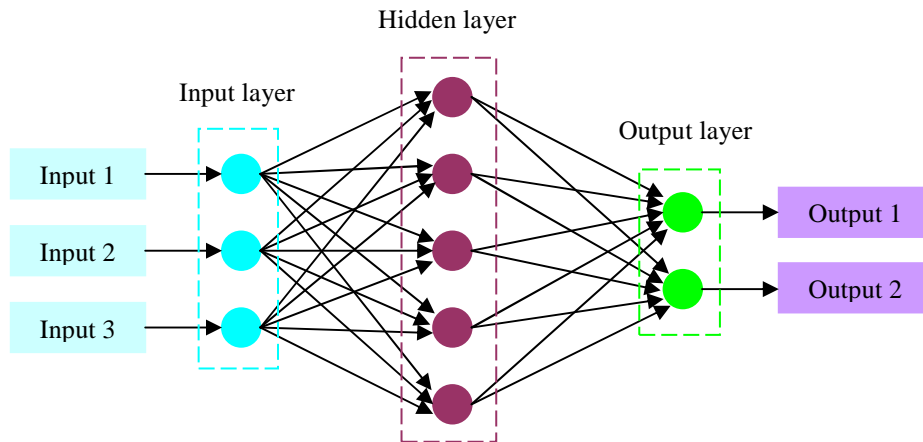


Figure B-3: Example of Multi-Layer Feed-Forward Non-Recurrent NN

Like a biological neuron, a node receives input data from outside, processes the data, and sends out an output datum. The working process of a generic node is shown in Figure B-4. This node has 2 inputs $x_1 = 1$ and $x_2 = 2$; the inputs are weighted by the weight factors $w_1 = 1$ and $w_2 = 2$, respectively; in addition, it has a bias $b = 1$ and a

corresponding weight factor $w_0 = 1$; since the sum of the weighted values including the bias term, i.e. $w_0b + w_1x_1 + w_2x_2 = 6$, is greater than the threshold value $\Theta = 4$, which has to be reached or exceeded for the node to produce a reaction (the neuron “fires”), a reaction $R = 6$ is obtained; then the reaction R is processed by a transfer function f , which is a hard limiter function in this example; finally, this node produce an output $o = 1$.

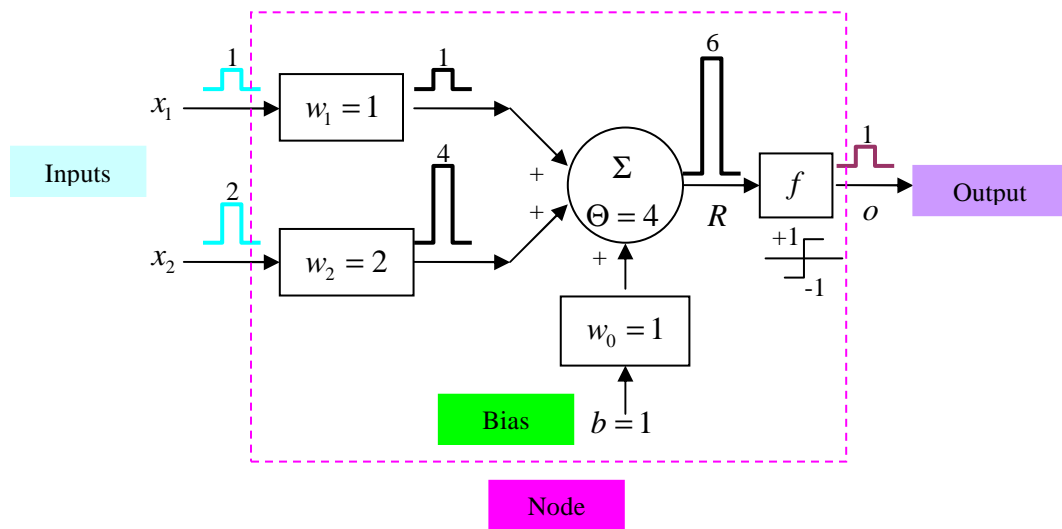


Figure B-4: Working Process of a Generic Node [119]

The nodes in different layers perform different processes. The nodes in the input layer directly pass the inputs to the nodes in the hidden layer, and each node has only one input with a unit weight factor. These nodes have no bias, threshold value, or transfer function. The nodes in the hidden layer usually perform the whole process shown in Figure B-4, except that usually the threshold value is not used (i.e. set to negative infinity). The nodes in the output layer have multiple inputs, and outputs of these output

layer nodes are the weighted sums of the inputs. These nodes have no threshold value, or transfer function.

One popular transfer function is the logistic sigmoid function as below, which always outputs a value between 0 and 1.

$$S(R) = \frac{1}{1 + e^{-R}}$$

The number of nodes in the input layer is the same as the number of design variables, and the number of nodes in the output layer is the same as the number of responses. It has been shown in Ref. [120] that an ANN can approximate any continuous function arbitrarily well as long as there are enough nodes in the hidden layer(s). Therefore, usually only one hidden layer is used and the number of nodes in this layer can be determined by the required accuracy level or other criteria that will be discussed later.

If only one hidden layer is used and the transfer function is the above sigmoid function, the output of a node in the hidden layer can be given as

$$H_j = S(R_j) = S\left(b_j + \sum_{i=1}^n w_{ij} x_i\right)$$

where H_j is the output of the j^{th} hidden node; R_j is the reaction of the j^{th} hidden node; b_j is the bias term for the j^{th} hidden node; w_{ij} is the weight factor for the i^{th} design variable; x_i is the i^{th} design variable; and n is the number of design variables.

Then the output of a node in the output layer, i.e. a response of this ANN can be given as

$$\hat{Y}_k = c_k + \sum_{j=1}^{N_H} d_{jk} H_j$$

where \hat{Y}_k is the k^{th} response; c_k is the bias term for the k^{th} response; d_{jk} is the weight factor for the j^{th} hidden node; H_j is the output of the j^{th} hidden node; and N_H is the number of hidden nodes.

Combine the above three equations together, a response of this ANN with the sigmoid function as the transfer function is given as

$$\hat{Y}_k = c_k + \sum_{j=1}^{N_H} d_{jk} \left[1 + \exp\left(-b_j - \sum_{i=1}^n w_{ij} x_i\right) \right]^{-1} \quad (\text{B. 8})$$

The process is called training or learning that estimates the weight factors and bias terms in Equation B.8. The two most common kinds of training processes are supervised and unsupervised training. If the ANN is trained to match the known values of the responses (target responses) for a given set of sample points, this is supervised training. Supervised training is used for surrogate-modeling. If there are no target responses to match, the weight factors and bias terms are adjusted according to certain given guidelines, this is unsupervised training. Unsupervised training is used for pattern recognition, classification, and control.

The training process itself is actually an optimization process. The optimization objective for supervised training is to minimize the model fitting errors between the target and predicted responses. There are many optimization algorithms that can be used for ANN training, such as gradient-based methods, Simulated Annealing, and Boltzmann machine [119]. The Matlab[®] NN toolbox has a variety of optimization algorithms for selection.

The main disadvantage is that ANN needs a large training sample, in terms of thousands, in order to obtain good accuracy. Additionally, there are two issues have to be considered when selecting an optimization algorithm. First, there is the probability that an algorithm becomes trapped in the local minima. If the problem is known to be multimodal, it is better to select a global algorithm, such as Simulated Annealing or Genetic Algorithm. Second, if the number of nodes in the hidden layer is also to be determined during the training process, a tradeoff should be carefully made between overfitting and underfitting. An overfitted ANN surrogate model usually will have good model fitting accuracy but bad model predicting accuracy. An underfitted ANN surrogate model is the opposite: bad model fitting accuracy but better predicting accuracy. The early stopping and regularization methods can be used for this tradeoff [37].

B.5 Gaussian Process

Gaussian Process has been applied to various problems in a large number of fields. It is a type of surrogate-modeling method that assumes that the (joint) distribution of the predicted values of the surrogate model at any points is a multivariate Gaussian distribution [121]. Two good summaries of the GP theory are provided in Ref. [40] and [122], and a realization of the GP theory is given in Ref. [37]. The GP realization given in Ref. [37] is as follows.

Given a sample $S_s : \{(y_1, X_1), (y_2, X_2), \dots, (y_s, X_s)\}$, assume a GP surrogate model is constructed based on this sample S_s such that the error term in Equation 2.14 follows an identical independent normal distribution $N(\mu, \sigma^2)$. Then the probability $P(Y_s | S_s, \Theta)$ follows a (joint) Gaussian distribution and is given by

$$P(Y_s | S_s, \Theta) = \frac{1}{\sqrt{(2\pi)^s |C_s|}} \exp\left[-\frac{1}{2}(Y_s - \mu)^T C_s^{-1} (Y_s - \mu)\right]$$

where $Y_s = [y_1, y_2, \dots, y_s]^T$, C_s is the covariance matrix for the sample S_s , μ is the mean, and Θ is the set of parameters in the covariance function and will be given later.

Then, when a new pair (y_{s+1}, X_{s+1}) is included, the probability $P(Y_{s+1} | S_{s+1})$ is similarly given by

$$P(Y_{s+1} | S_{s+1}, \Theta) = \frac{1}{\sqrt{(2\pi)^{s+1} |C_{s+1}|}} \exp\left[-\frac{1}{2}(Y_{s+1} - \mu)^T C_{s+1}^{-1} (Y_{s+1} - \mu)\right]$$

$$\text{where } C_{s+1} = \begin{bmatrix} C_s & M \\ M^T & \kappa \end{bmatrix}, \quad M = [C(X_1, X_{s+1}), \dots, C(X_s, X_{s+1})]^T, \quad \kappa = C(X_{s+1}, X_{s+1}),$$

$$Y_{s+1} = [y_1, y_2, \dots, y_s, y_{s+1}]^T, \text{ and } S_{s+1} : \{(y_1, X_1), (y_2, X_2), \dots, (y_s, X_s), (y_{s+1}, X_{s+1})\}$$

The probability of obtaining a single response y_{s+1} is given by

$$P(y_{s+1} | S_s, X_{s+1}, \Theta) = \frac{1}{\sqrt{(2\pi) \frac{|C_{s+1}|}{|C_s|}}} \exp\left[-\frac{(y_{s+1} - \hat{y}_{s+1})^2}{2\sigma_{\hat{y}_{s+1}}^2}\right]$$

where $\hat{y}_{s+1} = M^T C_s^{-1} Y_s$, and $\sigma_{\hat{y}_{s+1}}^2 = \kappa - M^T C_s^{-1} M$, i.e. the predicted response for the new point X_{s+1} and its variance.

The popular covariance function is given by

$$C(X_i, X_j) = \theta_1 \exp\left[-\frac{1}{2} \sum_{k=1}^n \frac{(x_{ik} - x_{jk})^2}{r_k^2}\right] + \theta_2 + \delta_{ij} \theta_3$$

where $\Theta : \{\theta_1, \theta_2, \theta_3, r_{k=1, \dots, n}\}$ are the parameters to be estimated based on the sample,

and $\delta_{ij} = 1$, if $i = j$; otherwise 0.

The posterior probability of the parameter set Θ based on the known sample S_s is obtained by Bayes' theorem

$$P(\Theta | S_s) = \frac{P(Y_s | S_s, \Theta)P(\Theta)}{P(Y_s | S_s)}$$

Usually the parameter set Θ is estimated by maximizing the natural logarithmic likelihood of its posterior probability.

$$\begin{aligned} L &= \ln P(\Theta | S_s) = \ln P(Y_s | S_s, \Theta) + \ln P(\Theta) - \ln P(Y_s | S_s) \\ &= \ln P(Y_s | S_s, \Theta) + \ln P(\Theta) + \text{const} \\ &= -\frac{1}{2} \ln |C_s| - \frac{1}{2} (Y_s - u)^T C_s (Y_s - u) - \frac{S}{2} \ln 2\pi + (\ln P(\Theta) + \text{const}) \end{aligned}$$

where $P(\Theta)$ is the prior knowledge about the probability of Θ , which can be either assumed to be a specific function of Θ or just ignored [123]; the term *const* comes from the fact that $P(Y_s | S_s)$ is independent of Θ although it is unknown and thus can also be ignored; and μ is assumed to be zero. Then, by maximizing L with respect to Θ for the sample S_s , the estimation of Θ is obtained. Then $\hat{y}_{s+1} = M^T C_s^{-1} Y_s$ is used to predict the response for the new point X_{s+1} .

B.6 Kriging

Kriging is a widely applied surrogate-modeling method originated from the mining and geostatistical fields [111, 60, 21, 124-127]. The Kriging model has two parts:

$$\hat{y}(X) = f(X) + Z(X) \quad (\text{B. 9})$$

where $\hat{y}(X)$ is the surrogate model function of interest, $f(X)$ is a function signifying the global behavior or tendency of the response, usually a polynomial function

$$f(X) = \sum_{j=1}^k \beta_j f_j(X)$$

of design variables with unknown yet coefficients β_j 's and known

monomials $f_j(X)$'s (for example, $f(X) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2$), and $Z(X)$ is a function that signifies the local variations concerned with, but not limited to, the immediate neighborhood, and is assumed to be a realization of a stochastic process with mean zero, variance σ^2 , and nonzero covariance. The covariance of $Z(X)$ between any two of the sample points X_i and X_j is given by:

$$\text{Cov}[Z(X_i), Z(X_j)] = \sigma^2 R(X_i, X_j) \quad (\text{B.10})$$

where $R(X_i, X_j)$ is the correlation function between the two sample points X_i and X_j . The correlation function is assumed by the user, and the most popular one is the Gaussian correlation function:

$$R(X_i, X_j) = \exp\left[-\sum_{k=1}^n \theta_k (x_{ik} - x_{jk})^2\right] \quad (\text{B.11})$$

where θ_k 's are the unknown parameters used to fit the model, x_{ik} and x_{jk} are the k^{th} components of the sample points X_i and X_j , respectively.

Let $Y_s = [y_1, y_2, \dots, y_s]^T$ as the responses of the sample points $\{X_1, X_2, \dots, X_s\}$, then the linear prediction equation is used for a new point X :

$$\hat{y}(X) = C^T(X)Y_s \quad (\text{B.12})$$

where $C(X)$ is a $s \times 1$ vector to be estimated through the sample (y_1, X_1) , (y_2, X_2) , ..., (y_s, X_s) . Actually, once the values of the coefficients β_k 's and the parameters θ_k 's are estimated through the sample, the estimation of $C(X)$ is obtained. However, the actual form of $C(X)$ does not need to be known as shown below.

Usually an optimizer such as Simulated Annealing is used with θ_k 's as its design variables, and the objective of the optimizer is to minimize the Akaike information criterion [124] (which will be discussed later) or maximize the likelihood function (see Ref. [115] for the definition) if $Z(X_j)$'s are normally and independently distributed with mean zero and variance σ^2 [128]. Of course, if θ_k 's are given a priori as many practices do (such as many examples in Ref. [124]), this optimizer is not needed.

For each set of values of θ_k 's given by the optimizer during the optimization process, the values of β_k 's are solved by minimizing the mean square error (MSE) as follows [21]:

$$\text{MSE}[\hat{y}(X)] = \text{E}[C^T(X)Y_s - Y(X)]^2$$

subject to the unbiasedness constraint:

$$\text{E}[C^T(X)Y_s] = \text{E}[Y(X)]$$

Let $ff(X) = [f_1(X), \dots, f_k(X)]_{1 \times k}$, $\beta = [\beta_1, \dots, \beta_k]_{1 \times k}^T$, $R = [R(X_i, X_j)]_{s \times s}$, $1 \leq i \leq s$,

$1 \leq j \leq s$, $F = \begin{bmatrix} ff(X_1) \\ \vdots \\ ff(X_s) \end{bmatrix}_{s \times k}$, and $r(X) = [R(X_1, X), \dots, R(X_s, X)]_{1 \times s}$, then without

knowing the actual form of $C(X)$ the estimations are obtained as [21]:

$$\hat{\beta} = (F^T R^{-1} F)^{-1} F^T R^{-1} Y_s, \text{ and } \hat{y}(X) = ff(X) \hat{\beta} + r(X) R^{-1} (Y_s - F \hat{\beta}) \quad (\text{B.13})$$

Alternatively, instead of the two-step method above, all the θ_k 's and β_k 's can be estimated by maximizing the likelihood function or minimizing the model predicting error using the cross validation method [125].

One major disadvantage of Kriging method is that the surrogate model construction can be very time-consuming if the sample size is large [60]. This is because the inverse of a large matrix R is by no means non-trivial and a n dimensional optimization problem has to be solved. Besides, the correlation matrix R can be singular if some sample points are close to each other.

Since the Kriging surrogate model has two parts, i.e. a polynomial for global behavior and a realization of a stochastic process for local variations, the Kriging method can be thought of a hybrid method of RSM and GP, if one expands the neighborhood for the second part to include all sample points over the entire design space [37]. However, Kriging does not make complete use of the Bayesian steps that the GP does.

If one focuses mainly on the first part, such as using high order polynomials for the first part, it could be argued that Kriging is an augmented RSM; on the other hand, if one focuses mainly on the second part, such as using first order polynomials or even a constant term as the first part, then Kriging is a method dominated by local behaviors. Therefore, Kriging can be tailored to resemble either of RSM and GP.

APPENDIX C: USING KARUSH-KUHN-TUCKER CONDITIONS TO CALCULATE b IN SVR

The Karush-Kuhn-Tucker (KKT) conditions [129] are necessary conditions for a point of a constrained optimization problem to be the optimal solution. See section 2.1 for the standard form of a constrained optimization problem, and see Equations 2.25 for the optimization problem for SVR. These conditions are used to calculate the intercept term b in SVR as follows.

$$\begin{aligned}
 & \underset{W, b, \xi_i^+, \xi_i^-}{\text{Minimize}} && \frac{1}{2} \|W\|^2 + C \sum_{i=1}^s (L(\xi_i^- + \varepsilon) + L(\xi_i^+ + \varepsilon)) \\
 & \text{subject to} && \begin{cases} y_i - \langle W, \Phi(X_i) \rangle - b \leq \varepsilon + \xi_i^+ \\ \langle W, \Phi(X_i) \rangle + b - y_i \leq \varepsilon + \xi_i^- \\ \xi_i^+, \xi_i^- \geq 0 \end{cases}
 \end{aligned} \tag{2.25}$$

Denote

$$Z = [W, b, \xi_i^+, \xi_i^-]^T, i = 1, \dots, s$$

$$F(Z) = \frac{1}{2} \|W\|^2 + C \sum_{i=1}^s (L(\xi_i^- + \varepsilon) + L(\xi_i^+ + \varepsilon))$$

$$g_i^+(Z) = y_i - \langle W, \Phi(X_i) \rangle - b - \varepsilon - \xi_i^+$$

$$g_i^-(Z) = -y_i + \langle W, \Phi(X_i) \rangle + b - \varepsilon - \xi_i^-$$

$$m_i^+(Z) = -\xi_i^+$$

$$m_i^-(Z) = -\xi_i^-$$

Then we get the first gradients

$$\nabla F(Z) = \begin{bmatrix} \partial_W F \\ \partial_b F \\ \partial_{\xi_i^+} F \\ \partial_{\xi_i^-} F \end{bmatrix} = \begin{bmatrix} W \\ 0 \\ C\partial L_{\xi_i^+}(\xi_i^+ + \varepsilon) \\ C\partial L_{\xi_i^-}(\xi_i^- + \varepsilon) \end{bmatrix}$$

$$\nabla g_i^+(Z) = \begin{bmatrix} -\Phi(X_i) \\ -1 \\ -1 \\ 0 \end{bmatrix}, \quad \nabla g_i^-(Z) = \begin{bmatrix} \Phi(X_i) \\ 1 \\ 0 \\ -1 \end{bmatrix}$$

$$\nabla m_i^+(Z) = \begin{bmatrix} 0 \\ 0 \\ -1 \\ 0 \end{bmatrix}, \quad \nabla m_i^-(Z) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1 \end{bmatrix}$$

The second KKT condition states that the product of a Lagrangian multiplier and an inequality constraint has to vanish. From this, we get

$$\alpha_i^+(\varepsilon + \xi_i^+ - y_i + \langle W, \Phi(X_i) \rangle + b) = 0 \quad (\text{C.1})$$

$$\alpha_i^-(\varepsilon + \xi_i^- + y_i - \langle W, \Phi(X_i) \rangle - b) = 0 \quad (\text{C.2})$$

$$\eta_i^+ \xi_i^+ = 0$$

$$\eta_i^- \xi_i^- = 0$$

The third KKT condition states that the grand sum of the first gradient of the objective function, the sum of the products of Lagrangian multiplier and first gradients of the inequality constraints, and the sum of the products of Lagrangian multiplier and first gradients of the equality constraints has to vanish. From this, we get:

$$\begin{bmatrix} W \\ 0 \\ C\partial L_{\xi_i^+}(\xi_i^+ + \varepsilon) \\ C\partial L_{\xi_i^-}(\xi_i^- + \varepsilon) \end{bmatrix} + \sum_{i=1}^s \alpha_i^+ \begin{bmatrix} -\Phi(X_i) \\ -1 \\ -1 \\ 0 \end{bmatrix} + \sum_{i=1}^s \alpha_i^- \begin{bmatrix} \Phi(X_i) \\ 1 \\ 0 \\ -1 \end{bmatrix} + \sum_{i=1}^s \eta_i^+ \begin{bmatrix} 0 \\ 0 \\ -1 \\ 0 \end{bmatrix} + \sum_{i=1}^s \eta_i^- \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1 \end{bmatrix} = 0 \quad (\text{C.3})$$

Subtract Equation C.2 from Equation C.1, and take sum of the differences over the index i , we get

$$\begin{aligned} & \varepsilon \sum_{i=1}^s (\alpha_i^+ - \alpha_i^-) + \sum_{i=1}^s (\alpha_i^+ \xi_i^+ - \alpha_i^- \xi_i^-) - \sum_{i=1}^s y_i (\alpha_i^+ + \alpha_i^-) + \\ & \sum_{i=1}^s (\alpha_i^+ - \alpha_i^-) \langle W, \Phi(X_i) \rangle + b \sum_{i=1}^s (\alpha_i^+ + \alpha_i^-) = 0 \end{aligned} \quad (\text{C.4})$$

From Equation C.3 we get

$$\begin{aligned} W - \sum_{i=1}^s (\alpha_i^+ - \alpha_i^-) \Phi(X_i) &= 0 \\ - \sum_{i=1}^s (\alpha_i^+ - \alpha_i^-) &= 0 \\ C\partial L_{\xi_i^+}(\xi_i^+ + \varepsilon) - \sum_{i=1}^s \alpha_i^+ - \sum_{i=1}^s \eta_i^+ &= 0 \\ C\partial L_{\xi_i^-}(\xi_i^- + \varepsilon) - \sum_{i=1}^s \alpha_i^- - \sum_{i=1}^s \eta_i^- &= 0 \end{aligned}$$

Thus we get

$$\begin{aligned} \bar{W} &= \sum_{i=1}^s (\alpha_i^+ - \alpha_i^-) \Phi(X_i) \\ \sum_{i=1}^s (\alpha_i^+ - \alpha_i^-) &= 0 \end{aligned} \quad (\text{C.5})$$

We can see that Equations C.5 are the same as Equations 2.28 and 2.29.

Substitute Equations C.5 into Equation C.4, and solve for b , we get

$$\begin{aligned}
\bar{b} &= \frac{1}{\sum_{i=1}^s (\alpha_i^+ + \alpha_i^-)} \left(\sum_{i=1}^s y_i (\alpha_i^+ + \alpha_i^-) - \sum_{i=1}^s (\alpha_i^+ - \alpha_i^-) \langle \bar{W}, \Phi(X_i) \rangle - \sum_{i=1}^s (\alpha_i^+ \xi_i^+ - \alpha_i^- \xi_i^-) \right) \\
&= \frac{1}{\sum_{i=1}^s (\alpha_i^+ + \alpha_i^-)} \left(\sum_{i=1}^s y_i (\alpha_i^+ + \alpha_i^-) - \sum_{i,j=1}^s (\alpha_i^+ - \alpha_i^-) (\alpha_j^+ - \alpha_j^-) k(X_i, X_j) - \sum_{i=1}^s (\alpha_i^+ \xi_i^+ - \alpha_i^- \xi_i^-) \right)
\end{aligned} \tag{C.6}$$

Substituting the results from Table 1 into Equation C.6, we get the results of b for different loss functions.

For quadratic loss function,

$$\bar{b} = \frac{1}{\sum_{i=1}^s (\alpha_i^+ + \alpha_i^-)} \left(\sum_{i=1}^s y_i (\alpha_i^+ + \alpha_i^-) - \sum_{i,j=1}^s (\alpha_i^+ - \alpha_i^-) (\alpha_j^+ - \alpha_j^-) k(X_i, X_j) - \frac{1}{2c} \sum_{i=1}^s ((\alpha_i^+)^2 - (\alpha_i^-)^2) \right)$$

For Laplace loss function,

$$\bar{b} = \frac{1}{\sum_{i=1}^s (\alpha_i^+ + \alpha_i^-)} \left(\sum_{i=1}^s y_i (\alpha_i^+ + \alpha_i^-) - \sum_{i,j=1}^s (\alpha_i^+ - \alpha_i^-) (\alpha_j^+ - \alpha_j^-) k(X_i, X_j) \right)$$

For ε -insensitive loss function,

$$\bar{b} = \frac{1}{\sum_{i=1}^s (\alpha_i^+ + \alpha_i^-)} \left(\sum_{i=1}^s y_i (\alpha_i^+ + \alpha_i^-) - \sum_{i,j=1}^s (\alpha_i^+ - \alpha_i^-) (\alpha_j^+ - \alpha_j^-) k(X_i, X_j) \right)$$

APPENDIX D: ERROR MEASURES AND TWO MODEL ASSESSMENT AND SELECTION METHODS

D.1 Model fitting Error and Predicting Error

The model fitting error is measured by the difference between the predicted response values by the surrogate model and the true response values of the sample points, while the model predicting error is measured by the difference between the predicted response values by the surrogate model and the true response values of the out-of-sample points. The predicting error is sometimes called model representing error or generalization error. Because the surrogate model is constructed from a given sample, the out-of-sample points are unknown to the surrogate model and thus are called “new” points to the surrogate model.

A good surrogate model should have both low model fitting error and low model predicting error, since the model fitting error is about the known sample and the model predicting error is about the (future) new points. However, there is no direct relationship between the model fitting error and model predicting error, i.e. a low model fitting error can not guarantee a low model predicting error [68]. The model predicting error has to be approximated since the number of out-of-sample points is infinite. Randomly generated new points can be used to reliably approximate the model predicting error, but this increases the cost to run the time-consuming physics-based models. The re-sampling methods can be used to approximate the model predicting error without using new points.

Of all the existing error measures the root mean square error (RMSE) and maximum absolute error (MAE) are most popular, shown in Equations D.1 and D.2,

respectively. Those error measures can be used to measure both the model fitting error and model predicting error. RMSE measures the overall approximation error, while MAE measures the local approximation error revealing regional areas of poor approximation [35]. Obviously, RMSE and MAE are non-parametric estimations of the error, since calculation of these two measures does not need to know the distribution of the errors.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n_{error}} (y_i - \hat{y}_i)^2}{n_{error}}} \quad (\text{D. 1})$$

$$\text{MAE} = \max |y_i - \hat{y}_i|, \quad i = 1, \dots, n_{error} \quad (\text{D. 2})$$

where n_{error} is the number of points used to calculate the error, including sample points and/or new points.

In this research, only the RMSE is used because the overall accuracy is of more concern in surrogate-modeling for engineering problems.

D.2 Fundamental of the Re-Sampling Methods: the Jackknife Method

The jackknife method and the re-sampling methods such as cross validation and bootstrap are originally statistical inference methods to estimate statistics (statistical parameters). Later on the re-sampling methods are applied to model assessment and model selection problems. Before introducing the re-sampling methods, the theory of the jackknife is introduced since it is the foundation of the re-sampling methods.

Suppose a single random variable X with an unknown distribution F , and a sample $S : \{X_1, X_2, \dots, X_s\}$ of size s . Thus $X_1, X_2, \dots, X_s \stackrel{iid}{\sim} F$. It has been well known that the expectation EX of this random variable X can be estimated by the sample

average \bar{x} from the observed sample $\{X_1 = x_1, X_2 = x_2, \dots, X_s = x_s\}$ according to the Law of Large Number:

$$\bar{x} = \sum_{i=1}^s \frac{1}{s} x_i = \frac{1}{s} \sum_{i=1}^s x_i \quad (\text{D. 3})$$

In a general sense, \bar{x} is the estimation of the statistic EX as a function of the sample $\{X_1, X_2, \dots, X_s\}$, i.e. $\bar{x} = \bar{x}(X_1, X_2, \dots, X_s)$. However, the Equation D.3 can not be extended in any obvious way to a general statistic θ other than EX , e.g. the sample median and RMSE, to estimate its expectation $E\theta$. Instead, the jackknife method can be used to make this extension [130].

Denote $\bar{\theta}$ as the estimation of the expectation $E\theta$, and $\hat{\theta}$ as the estimation of the statistic θ from the sample $\{X_1, X_2, \dots, X_s\}$. Thus $\hat{\theta} = \hat{\theta}(X_1, X_2, \dots, X_s)$. Further, denote $\hat{\theta}_{(k)}$ as an estimation of θ from a sub-sample deleting the k^{th} sample point X_k , i.e.

$$\hat{\theta}_{(k)} = \hat{\theta}(X_1, X_2, \dots, X_{k-1}, X_{k+1}, \dots, X_s) \quad (\text{D. 4})$$

Then the estimation of the expectation $E\theta$ is given as

$$\bar{\theta} = \sum_{i=1}^s \frac{1}{s} \hat{\theta}_{(k)} = \frac{1}{s} \sum_{i=1}^s \hat{\theta}_{(k)} \quad (\text{D. 5})$$

It can be seen that re-sampling methods, such as cross validation and bootstrap, are similar to the above jackknife method. In fact, the re-sampling methods and the jackknife method are closely connected in theory [130], and the theory of the jackknife method provides the foundation for all the re-sampling methods.

Given an error measure such as RMSE, the problem of model selection with the jackknife method can be stated as: select the model with the minimum expected error measure, where the expected error measure is estimated by Equation D.5, because the

error measure is one statistic θ . In this case, the meaning of Equation D.4 is to calculate the error measure from a surrogate model that is fitted from the sub-sample $S_{(k)} : \{(y_1, X_1), (y_2, X_2), \dots, (y_{k-1}, X_{k-1}), (y_{k+1}, X_{k+1}), \dots, (y_s, X_s)\}$. Since the expected error measure considers future points, i.e. the points X_k 's that are left out, it is a measure of the model predicting error.

D.3 The Cross Validation Method

The cross validation method is used to estimate the model predicting error. When applied to model selection, cross validation is a re-sampling method for model selection according to the predicting error of the candidate surrogate models [131]. The surrogate model with the minimum expected predicting error will be selected. The RMSE is a popular measure of the predicting error. The basic idea is to split the sample of size s into two parts; the first part with size s_c is used to construct the surrogate model; the second part with size $s_v = s - s_c$ is used to assess the predicting error of the model (model validation); a new splitting of the sample is executed, and the aforementioned process is repeated; this process is repeated many times, and the expectation of the predicting error is estimated as the average of the predicting errors obtained with different sample splitting. Figure D-1 shows the general scheme of cross validation. From these procedures one can see that the cross validation method is similar to the jackknife method with respect to estimation of the expectation of the predicting error and creation of the two sub-samples during iteration.

Several approaches of the cross validation method are proposed with different ways to split the sample into two parts. Four of those approaches are summarized below.

The basic cross validation [130, 132] first randomizes the original sample; then splits the sample into two halves; the first half is used to construct the surrogate model, and the other half is used to assess the predicting error; then switch the two halves; and repeat the process.

The k -fold cross validation [131, 133] first randomizes the original sample; then splits the randomized original sample into k sub-samples of approximately equal size; then a model is constructed from $(k-1)$ sub-samples and the remaining one sub-sample is used to assess the model; this process is repeated k times, each time leaving out one different sub-sample.

The leave- k -out cross validation [132] first draws k sample pairs out of the original sample; then the remaining $(s-k)$ sample pairs are used to construct the model, and the left out k sample pairs are used to assess the model; this process is repeated for all $\binom{s}{k}$ ways of drawing k sample pairs. This approach is more computationally expensive than the k -fold cross validation.

The leave-one-out cross validation [132] is a special case of leave- k -out cross validation with $k=1$. Obviously, its computational expense is much less than that of the leave- k -out cross validation for $k > 1$.

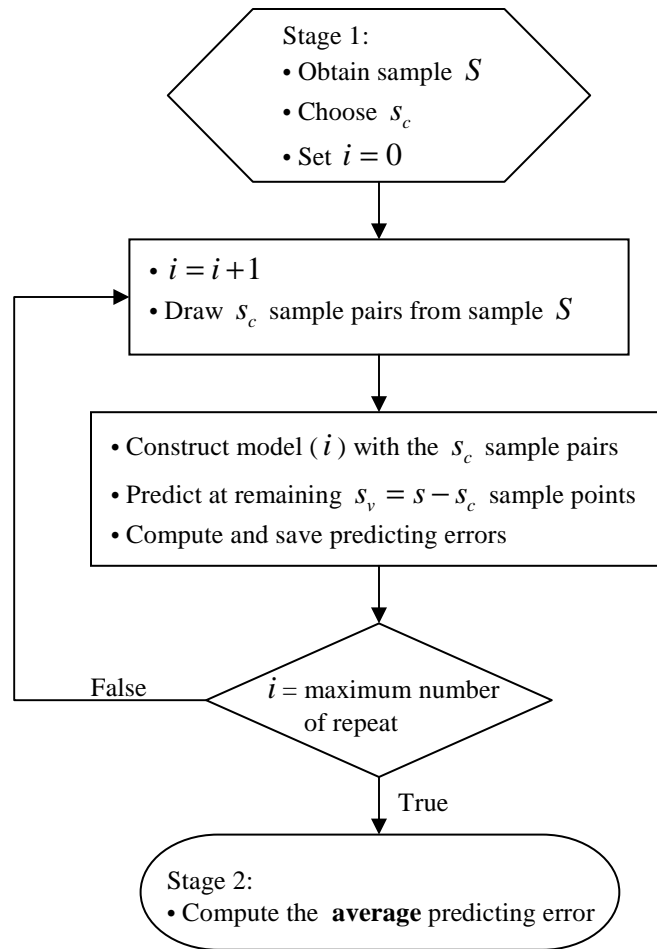


Figure D-1: General Scheme of Cross Validation for Model Selection

D.4 The Bootstrap Method

The bootstrap method is another method to estimate the model predicting error. When applied to model selection, bootstrap is one re-sampling method for model selection according to the predicting errors of the candidate surrogate models. There are many approaches of the bootstrap method with quite different procedures, such as Monte Carlo bootstrap [130, 134] and fast bootstrap [135], but the general procedures consist of drawing sample pairs with replacement within the original sample to obtain a bootstrap sample. Thus the size of the bootstrap sample is the same as that of the original sample. Since the sample pairs are draw with replacement, some sample pairs of the original

sample may appear in the bootstrap sample many times, while some may not at all. Then this bootstrap sample is used to construct a surrogate model and an estimated error measure (such as the model fitting error) is calculated for this bootstrap sample. This process is repeated B times. Then the average of the estimated error measures is used as the estimation of the expectation of the error measure.

From the above procedure of the bootstrap method one can see that the bootstrap method is similar to the cross validation method. There are three main differences between those two methods: the way to draw sample pairs, the size of the samples to construct the intermediate models, and the way to calculate the error measure. Two advantages of bootstrap are that its computational load is reduced from the k -fold cross validation and it results in lower variance for the estimated error measures [135].

APPENDIX E: THE CONCEPTS OF MPP AND LSF OF FPI

Let $Z = Z(X)$ being the response function of random variables X_1, X_2, \dots, X_n , and $g(X) = Z(X) - z_0$ being the limit state function (LSF), where z_0 is the critical value, and $g > 0$ is undesirable (failure).

The fast probability integration (FPI) methods are a family of methods used to estimate the probability P_f of achieving response values above the critical value z_0 , i.e. $g > 0$. This family includes the first order reliability method (FORM), second order reliability method (SORM), advanced FORM (AFORM), advanced mean value method (AMV), etc, and there are good descriptions of the FPI family methods in [83].

According to the Equation 2.53, the probability of $g > 0$ can be calculated as

$$P_f = P(g > 0) = \int_{R_g} \dots \int f_X(x_1, x_2, \dots, x_n) dx_1 dx_2 \dots dx_n$$

where R_g is the region over which $g(X) > 0$, and $f_X(x_1, x_2, \dots, x_n)$ is the joint PDF of random variables X_1, X_2, \dots, X_n . Therefore, the LSF $g(X) = 0$ “cut-off” a section of the joint PDF of the random variables X_1, X_2, \dots, X_n .

The FPI method uses the concept of most probably point (MPP) to estimate this probability P_f of violating the LSF. The MPP is the point at which the function $g(X) = 0$ circumscribes a contour line of the joint PDF, as shown at the left side of Figure B-1 of a bivariate example. The MPP can be found most conveniently in the transformed U-space in which all random variables are independently normally distributed. In the U-space, the MPP is the point on the transformed function $g(U) = 0$ that is closest to the origin, as shown at the right side of Figure B-1. The transformation

can be done using the Rosenblatt transformation method, and the distance of the MPP to the origin β is used to estimate P_f as $P_f = \Phi(-\beta)$ [83], where $\Phi(\cdot)$ is the CDF of the standard univariate normal distribution.

For the example shown in Figure E-1, $\Phi(-\beta)$ is the probability defined by the region at the hatched side of the straight **red** line, and this estimation of the probability P_f can be easily seen to be smaller than the real probability P_f . In general, the estimation $\Phi(-\beta)$ can be greater or smaller than the real probability P_f , and the difference between these two can be substantial.

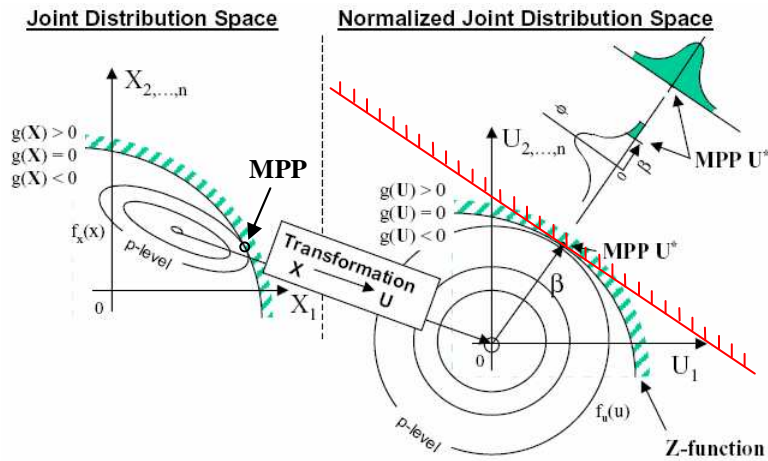


Figure E-1: Most Probable Point Location [136]

APPENDIX F: THE MULTI-VARIATE MONTE CARLO SAMPLING

Let $Y = Y(x_1, x_2, \dots, x_n)$ a function of the random variables X_1, X_2, \dots, X_n , and then Y is a random variable as well. Now one wants to generate sample points of Y through the sample points of X_1, X_2, \dots, X_n .

If the random variables X_1, X_2, \dots, X_n are mutually independent and the marginal PDF's of these variables are known, then the sample points of each random variable X_i can be generated as

$$x_i = F_i^{-1}(U) \quad (2.11)$$

where $F_i^{-1}(\cdot)$ is the inverse function of $F_i(x)$, $F_i(x)$ is the marginal CDF of X_i , and U is a uniform random variable of which values are generated by a (pseudo-) random number generator in computer experiments. Figure 2-4 shows the process to generate a sample point by univariate MC sampling.

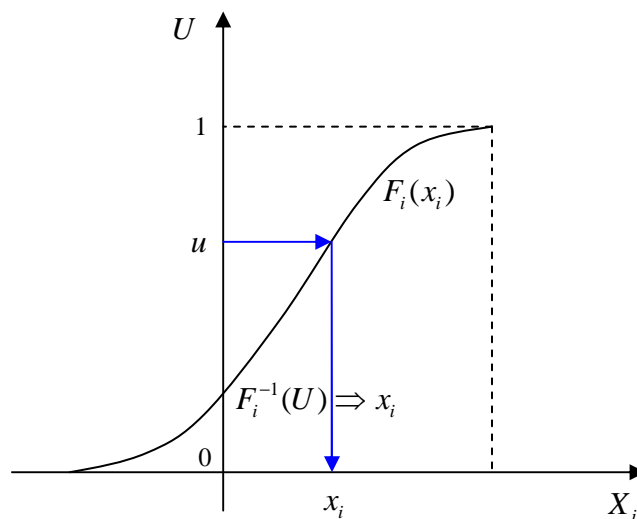


Figure 2-4: Univariate Monte Carlo Sampling Process

If the random variables X_1, X_2, \dots, X_n are dependent, suppose one knows the joint PDF $f_X(x_1, x_2, \dots, x_n)$ random variables X_1, X_2, \dots, X_n . First obtain the joint CDF $F_X(x_1, x_2, \dots, x_n)$. Generate the sample points of $(n-1)$ random variables using Equation 2.11. Substitute the sample points of the $(n-1)$ random variables into the joint CDF $F_X(x_1, x_2, \dots, x_n)$, and obtain the sample point of the last one random variable using Equation 2.11 and the joint CDF $F_X(x_1, x_2, \dots, x_n)$.

If the random variables are dependent, it is very difficult to generate the sample points since the generation of the joint PDF and CDF is not easy. Fortunately, for most engineering problems the (random) design variables are independent.

APPENDIX G: TRUNCATED NORMAL DISTRIBUTION

The (univariate) normal distribution has wide applications in various fields. It is used to describe bell-shaped distributions of single random variable, or the approximately bell-shaped distribution of a (random) response resulted from a large number of random variables via the Central Limit Theorem. However, the theory of the normal distribution assumes the range of the random variable is from $-\infty$ to $+\infty$, which is not the case for most real-world applications and thus may lead to large errors. Therefore, the truncated normal distribution should be used for the situations in which the range is finite of the random variable or a response of multiple random variables. Here the theory of the doubly truncated, univariate normal distribution is summarized as follows [137].

First define the PDF $f(X)$ of a normally-distributed random variable X as:

$$f(X) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{X-\mu}{\sigma}\right)^2}, \quad -\infty \leq X \leq +\infty \quad (\text{G.1})$$

Then the PDF $f_{DTN}(X)$ of doubly-truncated, normally-distributed random variable X is given as:

$$f_{DTN}(X) = \begin{cases} 0, & -\infty \leq X \leq X_L \\ \frac{f(X)}{\int_{X_L}^{X_R} f(X)dX}, & X_L \leq X \leq X_R \\ 0, & X_R \leq X \leq +\infty \end{cases} \quad (\text{G.2})$$

where $f(X)$ is defined in Equation G.1, X_L and X_R are the left and right limits of random variable X , respectively, see Figure G-1.

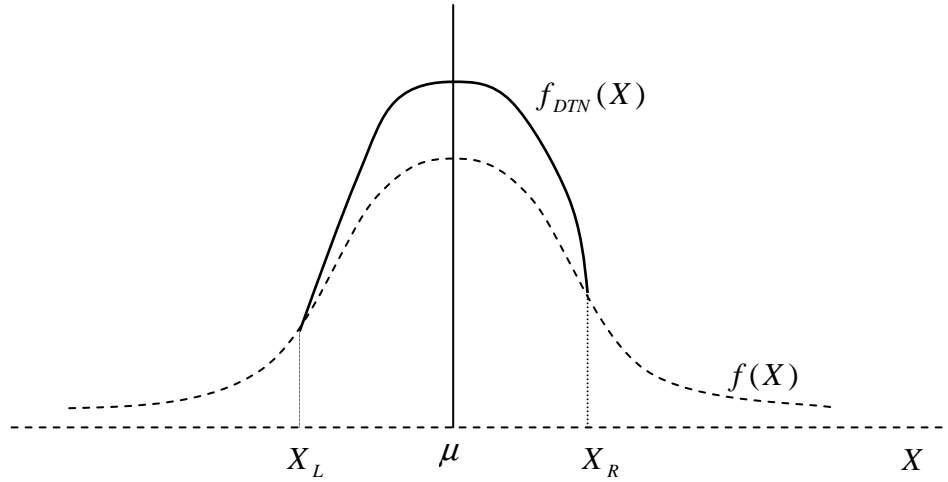


Figure G-1: Doubly-Truncated Normal Distribution (adapted from [137])

Denotes the CDF of $f(X)$ as $F(X)$, then the CDF of $f_{DTN}(X)$, $F_{DTN}(X)$, is given

as:

$$F_{DTN}(X) = \begin{cases} 0, & -\infty \leq X \leq X_L \\ \frac{F(X) - F(X_L)}{F(X_R) - F(X_L)}, & X_L \leq X \leq X_R \\ 0, & X_R \leq X \leq +\infty \end{cases} \quad (\text{G.3})$$

APPENDIX H: A TRANSPORT AIRCRAFT DESIGN OPTIMIZATION PROBLEM

This problem is based on the transport aircraft multidisciplinary design optimization problem in the class notes of Advanced Design Methods II, spring 2003, School of Aerospace Engineering, Georgia Institute of Technology, and is extended to a two-objective optimization problem under probabilistic constraints.

H1. Problem Description

The design of a transport aircraft is a complex, multidisciplinary process. Traditionally, the process is decomposed into aerospace engineering disciplines, such as aerodynamics, weights, performance, etc, in order to make the problem easier to manage. The closely coupled relationships and natural iteration between the various contributing analyses (CA's) provides an opportunity to use MDO techniques to improve the efficiency with which the design can be optimized. However, because of the uncertainties inevitably existing in the early stages of design, the design solutions can not be taken deterministically, but should be subject to probabilistic constraints.

Assume that the aircraft design team is decomposed into four disciplinary CA's:

CA	Discipline
D	Zero-lift Drag CA
A	Aerodynamics CA
W	Weights CA
P	Performance CA

The team's objective is to design a mid-range passenger jet transport so that the productivity index (PI) is maximized and the installed total engine thrust (T_i) is minimized. PI is a measure of the speed and cargo carrying capability of an aircraft normalized by the sum of its empty and fuel weights. T_i is considered to be a measure of both purchase price and operational cost. There are five primary design variables:

Variable	Name	Range of Mean	Distribution
b	wing span (ft)	[95 145]	3σ symmetrically truncated normal, $\sigma = \text{range}/2000$
l	fuselage length (ft)	[120 140]	3σ symmetrically truncated normal, $\sigma = \text{range}/2000$
S	wing area (ft ²)	[1300 1850]	3σ symmetrically truncated normal, $\sigma = \text{range}/2000$
W_{to}	takeoff gross weight (lb)	[155000 180000]	3σ symmetrically truncated normal, $\sigma = \text{range}/2000$
T_i	installed total engine thrust (lb)	[20000 35000]	3σ symmetrically truncated normal, $\sigma = \text{range}/2000$

And there are two coupling variables:

Variable	Name	Range*
V_{br}	best range cruise speed (ft/sec)	[550 800]
$W_{landing}$	landing gross weight (lb)	[75000 150000]

* For the purpose of surrogate model construction

The optimization problem under probabilistic constraints can be stated as follows:

$$\text{Maximize: } PI = \frac{V_{br} W_{pay}}{(1 - R_{fa} + R_{fr}) W_{to} - W_{pay} - W_{fix}}$$

$$\text{Minimize: } T_i$$

Subject to seven inequality probabilistic constraints:

$$P(S_{to} \leq 6000) \geq 0.85 \quad \text{takeoff field length (ft)}$$

$$P(S_l \leq 4000) \geq 0.85 \quad \text{landing field length (ft)}$$

$$P(U \geq 0.3) \geq 0.85 \quad \text{useful load fraction}$$

$$P(q_{to} \geq 2.7\%) \geq 0.85 \quad \text{takeoff climb gradient (one engine out)}$$

$$P(q_l \geq 2.4\%) \geq 0.85 \quad \text{aborted landing climb gradient (one engine out)}$$

$$P(\bar{R}_f \geq 1) \geq 0.85 \quad \text{overall mission fuel balance (available/required)}$$

$$P(AR \leq 10.5) \geq 0.85 \quad \text{wing aspect ratio}$$

H2. The Models of CA's

The models of CA's are given as some equations here to represent the real complex computer programs used by CA's.

Required Definitions and Constants

Variable	Value	Description
N_p	188	number of passengers

N	3	number of engines
b_t	1.944E-4 sec ⁻¹	engine specific fuel consumption
R	1.7632E7 ft	required range (approximately 2900 nmi)
h_{cruise}	35000 ft	cruise altitude
W_{pay}	30000 lb	payload (passengers and cargo)
W_{fix}	1100 lb	fixed equipment weight
ρ_{sl}	2.378E-3 slugs/ft ³	sea-level density
ν_{sl}	1.56E-4 ft ² /sec	sea-level kinematic viscosity
V_{to_l}	220 ft/sec	takeoff and landing speed
ρ_c	7.37 E-4 slugs/ft ³	cruise altitude density
ν_c	4.06E-4 ft ² /sec	cruise altitude kinematic viscosity
t/c	0.12	airfoil thickness-to-chord ratio
c_{l_max}	2.6	aircraft maximum lift coefficient (takeoff and landing)
c	S/b	mean aerodynamic wing chord (ft)
V_{br}	TBD	best range cruise speed (ft/sec)
L/D	Varies	lift-to-drag ratio at various flight conditions

D – Zero-Lift Drag Contributing Analysis

$\bar{S}_{wet} = 2$	wing wetted surface ratio
$d = 1.83(4.325 \frac{N_p}{l} + 1)$	fuselage diameter (to hold passengers) (ft)

$\bar{S}_s = \pi \frac{dl}{S} \left[1 - 2 \frac{d}{l} \right]^{\frac{2}{3}} \left[1 + \left(\frac{d}{l} \right)^2 \right]$	body wetted surface ratio
$c_{f_body_sl} = 0.455 \left[\log_{10} \left(\frac{V_{to_l} l}{v_{sl}} \right) \right]^{-2.58}$	body skin friction coefficient at sea level
$c_{f_wing_sl} = 0.455 \left[\log_{10} \left(\frac{V_{to_l} c}{v_{sl}} \right) \right]^{-2.58}$	wing skin friction coefficient at sea level
$c_{f_body_c} = 0.455 \left[\log_{10} \left(\frac{V_{br} l}{v_c} \right) \right]^{-2.58}$	body skin friction coefficient at cruise
$c_{f_wing_c} = 0.455 \left[\log_{10} \left(\frac{V_{br} c}{v_c} \right) \right]^{-2.58}$	wing skin friction coefficient at cruise
$\Delta C_{d0} = 0.005$	incremental drag coefficient
$(c_{d0})_{body_sl} = c_{f_body_sl} \left[1 + 0.0025 \left(\frac{l}{d} \right) + \frac{60}{\left(\frac{l}{d} \right)^3} \right] \bar{S}_s$	body drag contribution at sea level
$(c_{d0})_{wing_sl} = 1.1 c_{f_wing_sl} \left[1 + 1.2 \left(\frac{t}{c} \right) + 100 \left(\frac{t}{c} \right)^4 \right] \bar{S}_{wet}$	wing drag contribution at sea level
$(c_{d0})_{body_c} = c_{f_body_c} \left[1 + 0.0025 \left(\frac{l}{d} \right) + \frac{60}{\left(\frac{l}{d} \right)^3} \right] \bar{S}_s$	body drag contribution at cruise
$(c_{d0})_{wing_c} = 1.1 c_{f_wing_c} \left[1 + 1.2 \left(\frac{t}{c} \right) + 100 \left(\frac{t}{c} \right)^4 \right] \bar{S}_{wet}$	wing drag contribution at cruise
$c_{d0_sl} = (c_{d0})_{wing_sl} + (c_{d0})_{body_sl} + \Delta c_{d0}$	zero lift drag coefficient at sea level

$c_{d0_c} = (c_{d0})_{wing_c} + (c_{d0})_{body_c} + \Delta c_{d0}$	zero lift drag coefficient at cruise
---	--------------------------------------

A – Aerodynamics Contributing Analysis

$AR = \frac{b^2}{S}$	wing aspect ratio
$e = 0.96 \left[1 - \left(\frac{d}{b} \right)^2 \right]$	Oswald span efficiency factor
$k = \frac{1}{e \pi AR}$	quadratic drag polar
$c_{l_takeoff} = \frac{2W_{to}}{\rho_{sl} V_{to_l}^2 S}$	takeoff lift coefficient
$c_{l_landing} = \frac{2W_{landing}}{\rho_{sl} V_{to_l}^2 S}$	landing lift coefficient
$\left(\frac{L}{D} \right)_{takeoff} = \frac{c_{l_takeoff}}{c_{d0_sl} + k c_{l_takeoff}^2}$	lift-to-drag ratio at takeoff
$\left(\frac{L}{D} \right)_{landing} = \frac{c_{l_landing}}{c_{d0_sl} + k c_{l_landing}^2}$	lift-to-drag ration at landing
$\left(\frac{L}{D} \right)_{opt_cruise} = \frac{1}{2} \frac{1}{\sqrt{k c_{d0_c}}}$	optimum lift-to-drag ratio for cruise
$V_{br} = \sqrt{\frac{2W_{to}}{\rho_c S \sqrt{\frac{c_{d0_c}}{k}}}}$	best range cruise speed (ft/sec)

W – Weights Contributing Analysis

$\frac{W_f}{W_i} = \exp \left(\frac{-Rb_i}{V_{br} \left(\frac{L}{D} \right)_{opt_cruise}} \right)$	ratio of final to initial weight for cruise leg
$R_{fr} = 1.1 \left(1 - 0.95 \frac{W_f}{W_i} \right)$	required fuel fraction (with margin)
$\frac{W_{landing}}{W_{to}} = 1 - R_{fr}$	ratio of landing weight to takeoff weight
$\frac{W_{empty}}{W_{to}} = \frac{0.9592}{W_{to}^{0.0638}} + 0.38 \frac{T_i^{0.9881}}{W_{to}}$	ratio of empty weight to takeoff weight
$R_{fa} = 1 - \frac{W_{pay}}{W_{to}} - \frac{W_{fix}}{W_{to}} - \frac{W_{empty}}{W_{to}}$	available fuel fraction
$U = \frac{W_{pay}}{W_{to}} + R_{fr}$	useful load fraction

P – Performance Contributing Analysis

$q_{to} = \frac{1}{100} \left[\frac{T_i}{W_{to}} \frac{N-1}{N} - \frac{1}{\left(\frac{L}{D} \right)_{takeoff}} \right]$	takeoff climb gradient (one engine out)
$q_l = \frac{1}{100} \left[\frac{T_i}{W_{to}} \frac{N-1}{N} - \frac{1}{\left(\frac{L}{D} \right)_{takeoff}} \right]$	aborted landing climb gradient (one engine out)
$S_{to} = 20.9 \frac{W_{to}}{Sc_{l_max}} \frac{W_{to}}{T_i} + 87 \sqrt{\frac{W_{to}}{Sc_{l_max}}}$	takeoff field length (ft)

$S_l = 118 \frac{W_{landing}}{Sc_{l_max}} + 400$	landing field length (ft)
$\bar{R}_f = \frac{R_{fa}}{R_{fr}}$	overall mission fuel balance

APPENDIX I: A REUSABLE LAUNCH VEHICLE DESIGN OPTIMIZATION PROBLEM

This problem is based on the reusable launch vehicle (RLV) multidisciplinary design optimization problem in the class notes of Advanced Design Methods II, spring 2006, School of Aerospace Engineering, Georgia Institute of Technology, and is extended to a two-objective optimization problem under probabilistic constraints.

II. Problem Description:

Assume that the RLV design team has been decomposed into five disciplinary CA's as follows:

CA	Discipline
P	Propulsion
T	Trajectory optimization
Wu	Weight estimation – upper stage
Wb	Dry weight estimation – booster stage
S	Sizing and scaling

The team's objective is to design a RLV and propulsive upper stage so that the gross mass of the RLV and the upper stage W_{gross} and the vacuum specific pulse Isp_{vac} are minimized. The RLV booster stage will carry the upper stage to some staging point, where the upper stage will separate and continue to orbit. The booster then coasts back to a landing site. W_{gross} is a measure of the purchase price. Isp_{vac} is considered to be a

measure of operational cost since the propellant mass is the major portion of the gross weight and lower vacuum Isp means cheaper fuel. W_{gross} and Isp_{vac} are conflicting with each other because minimizing W_{gross} requires maximizing Isp_{vac} in order to carry less propellant. There are five primary design variables:

Variable	Name	Range of Mean	Distribution
ϵ	Expansion ratio of the engine nozzle	[10 100]	0.5σ symmetrically truncated normal, $\sigma = \text{range} / 2000$
T_{W_0}	Ratio of takeoff thrust and gross takeoff weight	[1.2 1.5]	0.5σ symmetrically truncated normal, $\sigma = \text{range} / 2000$
r	Engine oxidizer/fuel ratio (by weight)	[4.0 7.8]	0.5σ symmetrically truncated normal, $\sigma = \text{range} / 2000$
W_S	Landed weight divided by wing planform area	[20 50]	0.5σ symmetrically truncated normal, $\sigma = \text{range} / 2000$
ΔV_{split}	Percentage of total ΔV allocated to the booster stage	[0.25 0.75]	0.5σ symmetrically truncated normal, $\sigma = \text{range} / 2000$

And there are four coupling variables:

Variable	Name	Range*
W_{gross}	The gross takeoff weight including the booster and upper stage	[200000 2200000]
S_{ref}	Booster wing planform area (ft ²)	[1300 7500]

W_{prop}	Booster main ascent propellant weight (lb)	[60000 2000000]
W_{land}	Booster landed weight (lb)	[60000 240000]

* For the purpose of surrogate model construction

The optimization problem under probabilistic constraints can be stated as follows:

$$\text{Minimize: } W_{gross}$$

$$\text{Minimize: } Isp_{vac}$$

Subject to one inequality probabilistic constraint:

$$P(p_e \geq 5 \text{ psi}) \geq 0.30 \quad * \quad \text{Limit on nozzle exit pressure to avoid flow separation}$$

* The reason that the required probability is so small is because this design problem is quite sensitive to the perturbation around the converged design solutions, i.e. a small perturbation to a converged design solution will easily result in a non-converged design combination. Although there is only one probabilistic constraint, this RLV design problem still can demonstrate the feasibility of the proposed framework since the number of probabilistic constraints makes no difference to the operation of the probability counting Equation 2.55.

12. The Models of CA's

The models of CA's are given as some equations here to represent the real complex computer programs used by CA's.

Required Definitions and Constants

Variable	Value	Description
W_{pay}	32.174 ft/s ²	Earth surface gravity constant
n	5	Number engines on booster
W_{pay}	30000 lb	Payload to orbit
p_{sl}	2116 psf	Sea-level ambient pressure
p_c	446400 psf	Rocket engine chamber pressure
ΔV_{flight}	25000 ft/s	Actual flight ΔV to be provided by RLV booster and upper stage
k_{wing}	6 lb/ft ²	Wing weight per unit planform area
k_{tank}	0.7 lb/ft ³	Tank weight per unit volume
k_{body}	0.05	Body structural weight as fraction of landed weight
k_{TPS}	0.04	Thermal protection system (TPS) weight as fraction of landed weight
k_{gear}	0.03	Landing gear
k_{subsys}	30000 lb	Fixed subsystems weight
K_{margin}	0.15	RLV dry weight margin
$ox_{density}$	71.2 lb/ ft ³	Density of liquid oxygen
$h2_{density}$	4.41 lb/ft ³	Density of liquid hydrogen
$k_{residual}$	0.005	Residual propellant as fraction of ascent propellant
I_{sp_vac}	420 s	Isp of upper stage engine
λ_{upper}	0.15	Structural mass fraction of upper stage

P – Propulsion Contributing Analysis

$C_{f_vac} = 1.5467 + 2.5 * 10^{-3} * \epsilon + 3.66 * 10^{-2} * r$ $+ 1.6 * 10^{-4} * \epsilon * r - 1.481 * 10^{-5} * \epsilon^2 - 1.48 * 10^{-3} * r^2$	Vacuum thrust coefficient (RSE – by Tim Kokan)
$Isp_{vac} = 401.864 + 0.4729 * \epsilon + 12.0959 * r$ $+ 1.896 * 10^{-2} * \epsilon * r - 2.41 * 10^{-3} * \epsilon^2 - 1.41333 * r^2$	Vacuum Isp (RSE)
$p_e = 8.5535 - 0.1988 * \epsilon + 0.7546 * r - 8.98 * 10^{-3} * \epsilon * r$ $+ 1.37 * 10^{-3} * \epsilon^2 + 2.815 * 10^{-2} * r^2$	Nozzle exit pressure (psi) (RSE)
$\frac{T_{sl}}{W_e} = 62.1329 - 0.2551 * \epsilon - 0.9446 * r - 4.02 * 10^{-3} * \epsilon * r$ $+ 7.0 * 10^{-4} * \epsilon^2 + 0.18319 * r^2$	Engine sea-level thrust to weight ratio (RSE)
$T_{sl} = W_{gross} * T - W_0$	Total engine sea-level thrust (all engines) (lb)
$A_t = \frac{T_{sl}}{p_c C_{f_vac} - p_{sl} \epsilon}$	Total engine throat area (all engines) (ft ²)
$A_e = \epsilon A_t$	Total engine exit area (all engines) (ft ²)
$T_{vac} = T_{sl} + A_e p_{sl}$	Total engine vacuum thrust (all engines) (lb)

Note: Propulsion calculations are all for the engines on the booster stage only.

T – Trajectory Optimization Contributing Analysis

$\begin{aligned} \Delta V_{loss} = & 5073 - 13.739 * A_e - 2.4 * 10^{-3} * T_{vac} \\ & + 3.77 * 10^{-3} * W_{gross} + 0.2559 * S_{ref} \\ & - 4.39 * 10^{-6} * A_e * T_{vac} + 2.45 * 10^{-6} * A_e * W_{gross} \\ & - 1.47 * 10^{-4} * A_e * S_{ref} - 2.05 * 10^{-9} * T_{vac} * W_{gross} \\ & + 4.03 * 10^{-8} * T_{vac} * S_{ref} - 8.43 * 10^{-8} * W_{gross} * S_{ref} \\ & + 7.48 * 10^{-2} * A_e^2 + 1.00 * 10^{-9} * T_{vac}^2 \\ & + 8.19 * 10^{-10} * W_{gross}^2 + 1.80 * 10^{-2} * S_{ref}^2 \end{aligned}$	Ascent velocity losses for booster stage (ft/s) (RSE – by Tim Kokan)
$\Delta V_{booster} = \Delta V_{flight} * \Delta V_{split} + \Delta V_{loss}$	Total ΔV required for booster stage (ft/s)
$MR_{booster} = \exp \left[\frac{\Delta V_{booster}}{g_c * Isp_{vac}} \right]$	Required booster mass ratio
$\Delta V_{upper} = \Delta V_{flight} * (1 - \Delta V_{split})$	Total ΔV required for upper stage (ft/s)
$MR_{upper} = \exp \left[\frac{\Delta V_{upper}}{g_c * Isp_{upper}} \right]$	Required upper-stage mass ratio

Note: Assuming no losses for the upper stage.

Wu – Weight Estimation (Upper Stage) Contributing Analysis

$W_{upper} = \frac{W_{pay}}{1 - \left(\frac{MR_{upper} - 1}{MR_{upper}} \right) \left(\frac{1}{1 - \lambda_{upper}} \right)}$	Gross weight of upper stage (lb)
---	----------------------------------

Note: In a more detailed design, this simple equation could be replaced by a series of more detailed performance and sizing relations.

Wb – Dry Weight Estimation (Booster Stage) Contributing Analysis

$d = \frac{r + 1}{\frac{r}{ox_density} + \frac{1}{h2_density}}$	Propellant bulk density (lb/ft ³)
$W_{wing} = k_wing * S_{ref}$	Wing weight (lb)
$W_{body} = k_tank * \frac{W_{prop}}{d} + k_body * W_{land}$	Body weight (lb)
$W_{TPS} = k_TPS * W_{land}$	TPS weight (lb)
$W_{engines} = T_W_0 * \frac{W_{gross}}{\left(\frac{T_{sl}}{W_e}\right)}$	Engines weight (all) (lb)
$W_{gear} = k_gear * W_{land}$	Landing gear weight (lb)
$W_{subsys} = k_subsys$	Fixed subsystems weight (lb)
$W_{margin} = k_margin * (W_{wing} + W_{body} + W_{TPS} + W_{engines} + W_{gear} + W_{subsys})$	Dry weight margin (lb)
$W_{dry_booster} = W_{wing} + W_{body} + W_{TPS} + W_{engines} + W_{gear} + W_{subsys} + W_{margin}$	Total dry weight – booster stage (lb)

S – Sizing and Scaling Contributing Analysis

$W_{stage} = \frac{W_{upper} + W_{dry_booster}}{1 - k_residual * (MR_{booster} - 1)}$	Booster weight just prior to staging (includes gross weight of upper stage) (lb)
$W_{gross} = W_{stage} * MR_{booster}$	Booster gross takeoff weight (includes weight of upper stage) (lb)

$W_{residual} = W_{stage} - W_{upper} - W_{dry_booster}$	Booster residual main propellant weight (lb)
$W_{prop} = W_{gross} - W_{stage} - W_{upper}$	Booster main ascent propellant weight (lb)
$W_{land} = W_{stage} - W_{upper}$	Booster landed weight (lb)
$S_{ref} = \frac{W_{land}}{W_S}$	Booster wing planform area (ft ²)

Note: Assuming that the landed booster weight is equal to the booster weight just after staging (neglecting small propellant usage during coast back to ground).

REFERENCES

- [1] Dieter, G. E., *Engineering Design: A Materials and Processing Approach*, 3rd Edition, McGraw-Hill, 2000.
- [2] Blumrich, J. F., "Design", *Science*, Vol. 168, No. 3939, 1970, pp. 1551.
- [3] Raymer, D. P., *Aircraft Design: A Conceptual Approach*, 3rd Edition, AIAA, Virginia, USA, 1999.
- [4] Mattson, C., Messac, A., "Pareto Frontier Based Concept Selection under Uncertainty, with Visualization", *Optimization and Engineering*, Vol. 6, 2005, pp. 85.
- [5] Cho, H., Monthly Aviation, Republic of Korea, No. 197, 2005.
- [6] Kirby, M. R., "A Methodology for Technology Identification, Evaluation and Selection in Conceptual and Preliminary Aircraft Design", Ph. D., School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, 2001.
- [7] Mavris, D. N., DeLaurentis, D. A., et al, "A Stochastic Approach to Multi-disciplinary Aircraft Analysis and Design", 1998, AIAA-98-0912.
- [8] Bandte, O., "A Probabilistic Multi-Criteria Decision Making Technique for Conceptual and Preliminary Aerospace Systems Design", Ph.D., School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, Georgia, 2000.
- [9] Giesing, J. P., Barthelemy, J. M., "A Summary of Industry MDO Applications and Needs", 1998, AIAA-98-4737.
- [10] Du, X., Chen, W., "Collaborative Reliability Analysis under the Framework of Multidisciplinary Systems Design", *Optimization and Engineering*, Vol. 6, 2005, pp. 63.
- [11] Ran, H., Mavris, D. N., "A Framework for Determination of the Weak Pareto Frontier Design Solutions under Probabilistic Constraints", 2006, AIAA-2006-6960.
- [12] Marx, W. J., Mavris, D. N., Schrage, D. P., "Effects of Alternative Wing Structural Concepts on High Speed Civil Transport Life Cycle Costs", 1996, AIAA-96-1381.
- [13] Mavris, D. N., Bandte, O., Schrage, D. P., "Application of Probabilistic Methods for the Determination of an Economically Robust HSCT Configuration", 1996, AIAA-96-4090.

- [14] Marler, R. T., Arora, J. S., "Survey of Multi-Objective Optimization Methods for Engineering", *Structural and Multidisciplinary Optimization*, Vol. 26, 2004, pp. 369.
- [15] Zang, T. A., Hemsch, M. J., et al, "Needs and Opportunities for Uncertainty-Based Multidisciplinary Design Methods for Aerospace Vehicles", *NASA report*, 2002, NASA/TM-2002-211462.
- [16] Simpson, T. W., Booker, A., et al, "Approximation Methods in Multidisciplinary Analysis and Optimization: A Panel Discussion", 2002, http://endo.sandia.gov/DAKOTA/papers/panel_review.pdf (cited Nov. 2004)
- [17] Sobieszczanski-Sobieski, J., Altus, T. D., et al, "Bi-Level Integrated System Synthesis (BLISS) for Concurrent and Distributed Processing", 2002, AIAA-2002-5409.
- [18] Kroo, I., Manning, V., "Collaborative Optimization: Status and Directions", 2000, AIAA-2000-4721.
- [19] Ye, K. Q., "Orthogonal Column Latin Hypercubes and Their Application in Computer Experiments", *Journal of the American Statistical Association*, Vol. 93, No. 444, 1998, pp. 1430.
- [20] Giunta, A. A., Wojtkiewicz, S. F. Jr., et al, "Overview of Modern Design of Experiments Methods for Computational Simulations", 2003, AIAA 2003-0649.
- [21] Sacks, J., Welch, W. J., et al, "Design and Analysis of Computer Experiments", *Statistical Science*, Vol. 4, No. 4, 1989, pp. 409.
- [22] Simpson, T. W., Mauery, T. M., et al, "Comparison of Response Surface and Kriging Models for Multidisciplinary Design Optimization", 1998, AIAA-98-4755.
- [23] Barros, P. A. J., Kirby, M., Mavris, D. N., "Impact of Sampling Technique Selection on the Creation of Response Surface Models", 2004, AIAA-2004-01-3134.
- [24] Mckay, M. D., Beckman, R. J., et al, "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code", *Technometrics*, Vol. 21, 1979, pp. 239.
- [25] Iman, R. L., Conover, W. J., "Small Sample Sensitivity Analysis Techniques for Computer Models, With an Application to Risk Assessment", *Communications in Statistics, Part A. Theory and Methods*, Vol. 17, 1980, pp. 1749.
- [26] Helton, J. C., Davis, F. J., "Latin Hypercube Sampling and the Propagation of Uncertainty in Analyses of Complex Systems", *Reliability Engineering and System Safety*, Vol. 81, 2003, pp. 23.

- [27] Kalagnanam, J. R., Diwekar, U. M., "An Efficient Sampling Technique for Off-Line Quality Control", *Technometrics*, Vol. 39, No. 3, 1997, pp. 308.
- [28] Simpson, T. W., Lin, D. K. J., Chen, W., "Sampling Strategies for Computer Experiments: Design and Analysis", *Journal of Structural and Multidisciplinary Optimization*, Vol. 23, No. 1, 2001, pp. 1.
- [29] Metropolis, N., Ulam, S., "The Monte Carlo Method", *Journal of the American Statistical Association*, Vol. 44, No. 247, 1949, pp. 335.
- [30] Glasserman, P., *Monte Carlo Methods in Financial Engineering*, Springer, New York, 2004.
- [31] Vose, D., *Risk Analysis*, Second edition, John Wiley & Sons, LTD, 2000.
- [32] Kleijnen, J. P. C., "A Comment on blanning's metamodel for sensitivity analysis: the regression metamodel in simulation", *Interfaces*, Vol. 5, 1975, pp. 21.
- [33] Box, G. E. P., Draper, N. R., *Empirical Model-Building and Response Surfaces*, John Wiley & Sons, 1986.
- [34] Simpson, T. W., Peplinski, J. D., et al, "Metamodels for Computer-Based Engineering Design: Survey and Recommendations", *Engineering with Computers*, Vol. 17, 2001, pp. 129.
- [35] Clarke, S. M., Griebisch, J. H., et al, "Analysis of Support Vector Regression for Approximation of Complex Engineering Analysis", *ASME Design Engineering Technical Conferences - Design Automation Conference*, 2003, DETC2003/DAC-48759.
- [36] Meckesheimer, M., Barton, R. R., et al, "Metamodeling of Combined Discrete/Continuous Responses", *AIAA Journal*, Vol. 39, No. 10, 2001, pp. 1950.
- [37] Daberkow, D. D., "A Formulation of Metamodel Implementation Processes for Complex Systems Design ", Ph. D., School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, Georgia, 2002.
- [38] Barthelemy, J. F., Sobieszcanski-Sobieski, J., "Optimum Sensitivity Derivatives of Objective Functions in Nonlinear Programming", *AIAA Journal*, Vol. 21, No. 6, 1983, pp. 913.
- [39] Renaud, J. E., Gabriele, G. A., "Approximation in Non-Hierarchic System Optimization", *AIAA Journal*, Vol. 32, No. 1, 1994, pp. 198.
- [40] Seeger, M., "Gaussian Processes for Machine Learning", 2004, <http://www.cs.berkeley.edu/~mseeger/papers/bayesgp-tut.pdf> (cited June, 2006)

- [41] Myers, R. H., Montgomery, D. C., *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, John Wiley & Sons, New York, 1995.
- [42] Mavris, D. N., Hayden, W. T., "Formulation of an IPPD Methodology for the Design of a Supersonic Business Jet", 1996, AIAA-96-5591.
- [43] Tai, J. C., Mavris, D. N., Schrage, D. P., "Comparative Assessment of High Speed Rotorcraft Concepts (HSRC): Reaction Driven Stopped Rotor/Wing and Variable Diameter Tiltrotor", AIAA-97-5548.
- [44] Vapnik, V., *Statistical Learning Theorey*, Springer-Verlag, New York, 1998.
- [45] Vapnik, V., *Estimation of Dependences Based on Empirical Data (in Russian)*, (English Translation: Springer-Verlage, New York, 1982), Nauka, Moscow, 1979.
- [46] Burges, C. J. C., "A Tutorial on Support Vector Machines for Pattern Recognition", *Data Mining and Knowledge Discovery*, Vol. 2, 1998, pp. 121.
- [47] Chu, W., "Bayesian Approach to Support Vector Machines", Doctor of Philosophy, Department of Mechanical Engineering, National University of Singapore, 2003.
- [48] Gunn, S. R., "Support Vector Machines for Classification and Regression", 1998, <http://www.ecs.soton.ac.uk/~srg/publications/pdf/SVM.pdf> (cited Dec. 2003)
- [49] Fan, H., Dulikravich, G. S., et al, "Aerodynamic data modeling using support vector machines", *Inverse Problems in Science & Engineering (also presented as AIAA-2004-280)*, Vol. 13, No. 3, 2005, pp. 261.
- [50] Mukherjee, S., Osuna, E., et al, "Nonlinear Prediction of Chaotic Time Series Using a Support Vector Machine", *Proceedings of the IEEE Workshop in Neural Networks for Signal Processing*, Vol. 7, 1997, pp. 511-519.
- [51] Muller, K. R., Smola, A., et al, "Predicting Time Series with Support Vector Machines", *Proceedings of International conference on Artificial Neural Networks*, 1997, pp. 999.
- [52] Yang, H., "Margin Variations in Support Vector Regression for the Stock Market Prediction", Master of Philosophy, Department of Computer Science & Engineering, The Chinese University of Hong Kong, 2003.
- [53] Pai, P.-F., Hong, W. -Ch., "Support Vector Machines with Simulated Annealing Algorithms in Electricity Load Forecasting", *Energy Conversion and Management*, Vol. 46, 2005, pp. 2669.
- [54] Awad, M., Khan, L., "Applications and Limitations of Support Vector Machines", http://www.utdallas.edu/~maa013600/publications/DM_Encyclopedia_SVM_04.PDF (cited Jan. 2004)

- [55] Vapnik, V. N., *The Nature of Statistical Learning Theory*, Springer, 1995.
- [56] Chu, W., Keerthi, S. S., et al, "A Unified Loss Function in Bayesian Framework for Support Vector Regression", <http://guppy.mpe.nus.edu.sg/~chuvei/paper/icml.pdf> (cited Jan. 2004)
- [57] Smola, A. J., Scholkopf, B., "A Tutorial on Support Vector Regression", *Statistics and Computing*, Vol. 14, 2004, pp. 199.
- [58] Smola, A. J., Scholkopf, B., et al, "The Connection Between Regularization Operators and Support Vector Kernels", *Neural Networks*, Vol. 11, No. 4, 1998, pp. 637.
- [59] Fletcher, R., *Practical Methods of Optimization*, John Wiley & Sons, New York, 1987.
- [60] Jin, R., Chen, W., et al, "Comparative Studies of Metamodeling Techniques under Multiple Modeling Criteria", *Structural Multidisciplinary Optimization*, Vol. 23, 2001, pp. 1.
- [61] Haykin, S., *Neural Network: A Comprehensive Foundation*, Prentice Hall, 1999.
- [62] Hansen, M. H., Yu, B., "Model Selection and the Principle of Minimum Description Length", *Journal of the American Statistical Association*, Vol. 96, No. 454, 2001, pp. 746.
- [63] Akaike, H., "A New Look at the Statistical Model Identification", *IEEE Transactions on Automatic Control*, Vol. AC-19, No. 6, 1974, pp. 716.
- [64] Kullback, S., *Information Theory and Statistics*, Wiley, New York, 1959.
- [65] McQuarrie, A. D. R., Tsai, C. -L., *Regression and Time Series Model Selection*, World Scientific Publishing Co., Ltd., 1998.
- [66] Yen, J., Wang, L., "Application of Statistical Information Criteria for Optimal Fuzzy Model Construction", *IEEE Transactions on Fuzzy Systems*, Vol. 6, No. 3, 1998, pp. 362.
- [67] Schwarz, G., "Estimating the Dimension of a Model", *The Annals of Statistics*, Vol. 6, No. 2, 1978, pp. 461.
- [68] Qi, M., Zhang, G. P., "An Investigation of Model Selection Criteria for Neural Network Time Series Forecasting", *European Journal of Operational Research*, Vol. 132, 2001, pp. 666.
- [69] Waele, S. d., Broersen, P. M. T., "Order Selection for Vector Autoregressive Models", *IEEE Transactions on Signal Processing*, Vol. 51, NO. 2, 2003, pp. 427-433.

- [70] Kadilar, C., Erdemir, C., "Modification of the Akaike Information Criterion to Account for Seasonal Effects", *Journal of Statistical Computation and Simulation*, Vol. 73, No. 2, 2003, pp. 135.
- [71] Ustun, B., Melssen, W. J., et al, "Determination of Optimal Support Vector Regression Parameters by Genetic Algorithms and Simplex Optimization", *Analytica Chimica Acta*, Vol. 544, 2005, pp. 292.
- [72] Scholkopf, B., Bartlett, P., et al, *Shrinking the Tube: A New Support Vector Regression Algorithm*, Advances in Neural Information Processing Systems, edited by Kearns, M. S., Smola, A. J., MIT Press, Cambridge, 1999.
- [73] Cherkassky, V., Ma, Y., "Practical Selection of SVM Parameters and Noise Estimation for SVM Regression", *Neural Networks*, Vol. 17, 2004, pp. 113.
- [74] Alvin, K. F., Oberkampf, W. L., et al, "Methodology for Characterizing Modeling and Discretization Uncertainties in Computational Simulation", Sandia National Laboratories, Report No. SAND2000-0515, 2000.
- [75] Huysse, L., "Solving Problems of Optimization Under Uncertainty as Statistical Decision Problems", 2001, AIAA-2001-1519.
- [76] Dubi, A., *Monte Carlo Applications in Systems Engineering*, John Wiley & Sons, New York, 2000.
- [77] Shooman, M. L., *Probabilistic Reliability: an Engineering Approach*, 2nd edition, Krieger Publication Co., 1990.
- [78] Frits, A. P., "Formulation of an Integrated Robust Design and Tactics Optimization Process for Undersea Weapon Systems", Ph. D., School of Aerospace Engineering, Georgia Institute of Technology, 2004.
- [79] Koch, P. N., Yang, R. J., et al, "Design for six sigma through robust optimization", *Structural and Multidisciplinary Optimization*, Vol. 26, No. 3, 2004, pp. 235.
- [80] Wallace, J. M., "A Framework for Conducting Mechanistic Based Reliability Assessments of Components Operating in Complex Systems", Ph. D., School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, 2003.
- [81] Jin, R., Du, X. et al, "The Use of Metamodeling Techniques for Optimization under Uncertainty", *Structural Multidisciplinary Optimization*, Vol. 25, 2003, pp. 99.
- [82] Melchers, R. E., *Structural Reliability Analysis and Prediction*, 2nd edition, John Wiley & Sons, 1999.
- [83] Wu, Y. T., *FPI User's and Theoretical Manuals*, Southwest Research Institute, San Antonio, Texas, 1995.

- [84] Koch, P. N., Simpson, T. W., et al, "Statistical Approximations for Multidisciplinary Design Optimization: The Problem of Size", *JOURNAL OF AIRCRAFT*, Vol. 36, No. 1, 1999, pp. 275.
- [85] Gumbert, C. R., Newman, P. A., et al, "Effect of Random Geometric Uncertainty on the Computational Design of a 3-D Flexible Wing", 2002, AIAA-2002-2806.
- [86] Li, W., Padula, S., "Performance Trades Study for Robust Airfoil Shape Optimization", 2003, AIAA-2003-3790.
- [87] Oakley, D. R., Sues, R. H., et al, "Performance Optimization of Multidisciplinary Mechanical Systems Subject to Uncertainties", *Probabilistic Engineering Mechanics*, Vol. 13, No. 1, 1998, pp. 15.
- [88] Sues, R. H., Cesare, M. A., et al, "Reliability-Based Optimization Considering Manufacturing and Operational Uncertainties", *Journal of Aerospace Engineering*, Vol. 14, No. 4, 2001, pp. 166.
- [89] Lyle, K. H., Padula, S. L., "Application of Probabilistic Analysis to Aircraft Impact Dynamics", 2003, AIAA-2003-1482.
- [90] ChanKong, V., Haimes, Y. Y., *Multiobjective Decision Making: Theory and Methodology*, Elsevier Science Publishing Co., Inc., 1983.
- [91] Messac, A., Ismail-Yahaya, A., Mattson, C.A., "The Normalized Normal Constraint Method for Generating the Pareto Frontier", *Structural and Multidisciplinary Optimization*, Vol. 25, 2003, pp. 86.
- [92] Messac, A., Mattson, C. A., "Normal Constraint Method with Guarantee of Even Representation of Complete Pareto Frontier", *AIAA Journal*, Vol. 42, No. 10, 2004, pp 2101.
- [93] Ledsinger, L. A., "Solutions to Decomposed Branching Trajectories with Powered Flyback Using Multidisciplinary Design Optimization", Ph.D., School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, Georgia, 2000.
- [94] Brown, N., "Evaluation of Multidisciplinary Optimization (MDO) Techniques Applied to a Reusable Launch Vehicle", Master of Aerospace Engineering, School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, Georgia, 2004.
- [95] DeMiguel, A.-V., Murray, W., "An Analysis of Collaborative Optimization Methods", 2000, AIAA-2000-4720.
- [96] Smaling, R. M., Weck, O. L. de, "Fuzzy Pareto Frontiers in Multidisciplinary System Architecture Analysis", 2004, AIAA-2004-4553.
- [97] Sues, R. H., Oakley, D. R., Rhodes, G. S., "MDO of Aeropropulsion Components Considering Uncertainty", 1996, AIAA-96-4062.

- [98] Varadarajan, S., Chen, W. et al, "Robust Concept Exploration of Propulsion Systems with Enhanced Model Approximation Capabilities", *Engineering Optimization*, Vol. 32, No. 3, 2000, pp. 309.
- [99] Curry, B., Morgan, P. H., "Model Selection in Neural Networks: Some Difficulties", *European Journal of Operational Research*, 2004, Accepted.
- [100] McCullers, L. A., *FLOPS User's Guide*, Ver. 5.94, NASA Langley Research Center, Hampton, Virginia, 1998.
- [101] Johnson, C., Schutte, J., "Basic Regression Analysis for Integrated Neural Networks (BRAINN) Documentation", Aerospace Systems Design Laboratory, Georgia Institute of Technology, 2006.
- [102] Hwang, C.-L., Yoon, K., *Multiple Attribute Decision Making: Methods and Applications: A State-of-the-Art Survey*, Springer-Verlag, Berlin; New York, 1981.
- [103] Agrawal, G., Lewis, K., et al, "Intuitive Visualization of Pareto Frontier for Multi-Objective Optimization in n-Dimensional Performance", 2004, AIAA-2004-4434.
- [104] Agrawal, G., "The Hyperspace Pareto Frontier for Intuitive Visualization of Multiobjective Optimization Problems", Ph.D., State University of New York at Buffalo, 2005.
- [105] Wang, G. G., "Adaptive Response Surface Method Using Inherited Latin Hypercube Design Points", *Journal of Mechanical Design*, Vol. 125, 2003, pp. 210.
- [106] Bates, S. J., Sienz, J., et al, "Formulation of the Optimal Latin Hypercube Design of Experiments Using a Permutation Genetic Algorithm", 2004, AIAA-2004-2011.
- [107] Cioppa, T. M., "Efficient Nearly Orthogonal and Space-filling Experimental Designs for High-Dimensional Complex Models", Doctor of Philosophy, Naval Postgraduate School, 2002.
- [108] Fang, K. T., Lin, D. K., et al, "Uniform Design: Theory and Application", *Technometrics*, Vol. 42, No. 3, 2000, pp. 237.
- [109] Perez, V. M., Renaud, J. E., "Adaptive Experimental Design for Construction of Response Surface Approximation", 2001, AIAA-2001-1622.
- [110] Montgomery, D. C., *Design and Analysis of Experiments*, 4th Edition, Wiley & Sons, New York, 1997.
- [111] Jin, R., Sudjianto, A., "On Sequential Sampling for Global Metamodeling in Engineering Design", *Proceedings of DETC*, 2002, DETC2002/DAC-34092.
- [112] Owen, A. B., "Orthogonal Arrays for Computer Experiments, Integration and Visualization", *Statistica Sinica*, Vol. 2, 1992, pp. 439.

- [113] Fang, K. T., "Experimental design by Uniform Distribution", *Acta Mathematicae Applicatae Sinica*, Vol. 3, 1980, 363.
- [114] Fang, K. T., Ma, Ch. X. et al, "Centered L2-Discrepancy of Random Sampling and Latin Hypercube Design, and Construction of Uniform Designs", *MATHEMATICS OF COMPUTATION*, Vol. 71, 2000, pp. 275.
- [115] Hayter, A. J., *Probability and Statistics for Engineers and Scientists*, 2nd Edition, Duxbury, 2002.
- [116] Graybill, F. A., Iyer, H. K., *Regression Analysis: Concepts and Applications*, Duxbury Press, 1994.
- [117] Mackay, D. J. C., *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press, 2005.
- [118] Chen, W., Varadarajan, S., "Integration of Design of Experiments and Artificial Neural Networks for Achieving Affordable Concurrent Design", 1997, AIAA-1997-1230.
- [119] Kartalopoulos, S. V., *Understanding Neural Networks and Fuzzy Logic*, IEEE Press, New York, 1996.
- [120] Chauvin, Y., Rumelhart, D. E., *Back Propagation: Theory, Architectures, and Applications*, Lawrence Erlbaum Associates, 1994.
- [121] Neal, R. M., *Bayesian Learning for Neural Networks*, Springer-Verlag, New York, 1996.
- [122] Smola, A. J., Scholkopf, B., *Advanced Lectures on Machine Learning*, Springer-Verlag, Berlin, 2003.
- [123] Gibbs, M. N., "Bayesian Gaussian Processes for Regression and Classification", Ph. D., University of Cambridge, 1997.
- [124] Olea, R. A., *Geostatistics for Engineers and Earth Scientists*, Kluwer Academic Publishers, 1999.
- [125] Martin, J. D., Simpson, T. W., "A Study on the Use of Kriging Models to Approximate Deterministic Computer Models", ASME, 2003, DETC2003/DAC-48762.
- [126] Simpson, T. W., Mauery, T. M., et al, "Kriging Models for Global Approximation in Simulation-Based Multidisciplinary Design Optimization", *AIAA Journal*, Vol. 39, No. 12, 2001, pp. 2233.
- [127] Martin, J. D., Simpson, T. W., "Use of Adaptive Metamodeling for Design Optimization", 2002, AIAA-2002-5631.

- [128] Jones, D. R., Schonlau, M., et al, "Efficient Global Optimization of Expensive Black-Box Functions", *Journal of Global Optimization*, Vol. 13, 1998, pp. 455.
- [129] Vanderplaats, G. N., *Numerical Optimization Techniques for Engineering Design: with Applications*, 3rd Edition, Vanderplaats Research & Development Inc., 2001.
- [130] Efron, B., *The Jackknife, the Bootstrap and Other Resampling Plans*, Society for Industrial and Applied Mathematics, 1982.
- [131] Shao, J., "Linear Model Selection via Cross-Validation", *Journal of the American Statistical Association*, Vol. 88, No. 422, 1993, pp. 486.
- [132] Meckesheimer, M., Booker, A. J., et al, "Computationally Inexpensive Metamodel Assessment Strategies", *AIAA Journal*, Vol. 40, No. 10, 2002, pp. 2053.
- [133] Kohavi, R., "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection", *Proceedings of the 14th International Joint Conference on A. I.*, Vol. 2, Montreal, 1995, pp. 1137-1143.
- [134] Hinkley, *Bootstrap Methods and Their Application*, Cambridge University Press, 1997.
- [135] Lendasse, A., Simon, G., et al, "Fast Bootstrap Methodology for Regression Model Selection", *Neurocomputing*, Vol. 64, 2005, pp. 161.
- [136] Mavris, D. N., Mantis, G., et al, "Demonstration of a Probabilistic Technique for the Determination of Economic Viability", 1997, AIAA-97-5585.
- [137] Johnson, A. C., Thomopoulos, N. T., "Characteristics and Tables of the Doubly-Truncated Normal Distribution", 2002,
<http://www.stuart.iit.edu/faculty/workingpapers/thomopoulos/char-doubly.pdf>
(cited Aug. 2006)

VITA

Hongjun Ran was born in Chongqing, China on December 5, 1970. He spent most of his pre-college years in Shizhu County, Chongqing, China and graduated from the Shizhou Senior High School in 1988. He entered the Beijing University of Aeronautics and Astronautics, China in 1988 where he earned a Bachelor of Science degree with specialty in Aircraft Design in 1992 and a Master of Science degree with specialty in Aircraft Design in 1995. Then he worked as a mechanical engineer for 7 years for China Southern Airlines Co., Ltd, China. In the fall of 2002, he enrolled in the Ph.D. program at the Georgia Institute of Technology. Since then, he has received a Master of Science degree in Aerospace Engineering in 2003.