

STRATEGY FOR GLOBAL OPTIMIZATION AND POST-OPTIMALITY
USING LOCAL KRIGING APPROXIMATIONS

A Thesis

Submitted to the Graduate School
of the University of Notre Dame
in Partial Fulfillment of the Requirements
for the Degree of
Master of Science

by

Parveen K. Chandila, B.Tech (Hons.)

John E. Renaud, Director

Graduate Program in Aerospace and Mechanical Engineering

Notre Dame, Indiana

April 2004

STRATEGY FOR GLOBAL OPTIMIZATION AND POST-OPTIMALITY
USING LOCAL KRIGING APPROXIMATIONS

Abstract

by

Parveen K. Chandila

A strategy for global optimization based upon local approximate optimization has been developed in this research. The global design space is divided into local regions. Accurate Kriging approximations are constructed in the local regions to provide an inexpensive formulation of the system behavior. Each Kriging approximation is validated by using a cross-validation scheme. Gradient based local optimizers are used to identify local optima in each of the regions. A reduced trust region around the local optimum is further analyzed to provide the global optimum solutions. This approach possesses the capability of identifying global optimum solutions within reduced time frames as compared to existing methods.

A second investigation focused on the formulation and development of a non-linear post-optimality analysis. The non-linear response of the system is captured through local approximations. A cumulative approximation is constructed from the local response surface approximations via a blending function. The post-optimal solution is obtained by performing optimization over the cumulative approximation of the objective function and the constraints.

To my parents.

CONTENTS

FIGURES	vi
TABLES	viii
SYMBOLS	ix
ACKNOWLEDGMENTS	xi
CHAPTER 1: INTRODUCTION	1
1.1 Response Surface Approximation	1
1.2 Classification of Approximation	2
1.3 Optimization Categories	2
1.4 Research Problem Overview	3
1.4.1 Global Optimization	4
1.4.2 Non-Linear Post-Optimality Strategy	5
1.5 Overview of the Thesis	6
CHAPTER 2: KRIGING APPROXIMATION	8
2.1 Kriging Models	8
2.2 Mathematical Formulation	8
2.3 Illustration of Kriging Approximation	10
2.4 Features of Kriging Models	11
2.5 Disadvantages of Using Kriging Models	13
2.6 Model Validation	14
2.6.1 Measures of Error	15
2.6.2 Leave- <i>one</i> -out Cross-Validation Strategy	15
2.6.3 Approximate Cross Validation	16
2.7 Summary	18
CHAPTER 3: CUMULATIVE APPROXIMATION	19
3.1 Spline Interpolation for Cumulative Approximation	19
3.2 Drawbacks of Spline Interpolation	20
3.3 Cumulative Response Surface Approximation (CRSA)	20
3.4 Mathematical Formulation of CRSA	21
3.5 Illustration of CRSA	23
3.5.1 CRSA of Simple Test Function	23
3.5.2 CRSA of Complex Test Function	23
3.6 Summary	26

CHAPTER 4: GENETIC ALGORITHM	28
4.1 What are Genetic Algorithms?	28
4.2 Overview of GAs	29
4.3 GA Application Issues	29
4.3.1 Representation	29
4.3.2 Creation of Initial Population	30
4.3.3 Genetic Operators	31
4.3.4 Fitness Evaluation Function	36
4.4 Summary	36
 CHAPTER 5: GLOBAL OPTIMIZATION STRATEGY	 38
5.1 Survey of Global Optimization Methods	38
5.2 Pitfalls of Genetic Algorithms	39
5.3 Global Optimization Using Local Approximate Optimization (GO- LAO)	41
5.4 Illustration of the GO-LAO Algorithm	48
5.4.1 Division of Space	50
5.4.2 Constructing Accurate Approximations	51
5.5 Advantages of GO-LOA	55
5.6 Test Problems	56
5.6.1 Camelback Six-Hump Function	56
5.6.2 Engine Design Problem	66
5.6.3 Autonomous Underwater Vehicle (AUV) Design Problem	69
5.7 Summary	74
 CHAPTER 6: NON-LINEAR POST-OPTIMALITY ANALYSIS	 76
6.1 Post-Optimality Analysis	76
6.2 Linear Post-Optimality Strategy	77
6.3 Overview of Non-Linear Post-Optimality Analysis (NLPOA)	78
6.4 Mathematical Formulation	79
6.4.1 Cumulative Response Surface Approximation (CRSA)	80
6.4.2 Nonlinear Post-Optimality Analysis (NLPOA)	80
6.5 Advantages of NLPOA	82
6.6 Test Problems	83
6.6.1 Barnes Problem	83
6.6.2 High Performance Low Cost Structure (HPLC) Ten-bar Problem	87
6.6.3 Control-Augmented Structure Problem	91
6.7 Summary	92
 CHAPTER 7: CONCLUSIONS AND FUTURE WORK	 94
7.1 Summary	94
7.2 Conclusions	95
7.2.1 GO-LAO Approach	96
7.2.2 NLPOA Strategy	97
7.3 Recommendations for Future Work	98
7.3.1 Division of Global Space	98
7.3.2 Kriging Model Assessment	99

7.3.3	Improving Accuracy in Cumulative Approximations	100
APPENDIX A: TEST PROBLEMS		101
A.1	Camelback Six Hump Function	101
A.2	Engine Design Problem	101
A.3	Underwater Autonomous Vehicle Design Problem	105
	A.3.1 Dynamic Model Details	108
	A.3.2 Design Problem	108
A.4	Barnes Problem	110
A.5	High Performance Low Cost (HPLC) Structure Problem	112
A.6	Controls-Augmented Structure Problem	116
BIBLIOGRAPHY		119

FIGURES

2.1	Surface Plot of the Exact Camelback Function	11
2.2	Surface Plot of the Kriging Approximation of the Camelback Function	12
2.3	Extrapolation of a Kriging approximation of $y = \sin(x)$ in a wide region	13
3.1	Effect of Distance on the Blending Function	22
3.2	CRSA of the Simple Test Function	24
3.3	CRSA of the Complex Test Function	25
3.4	CRSA of the Complex Test Function With Additional Point	26
3.5	CRSA of the Complex Test Function With Another Additional Point	27
4.1	Flowchart of Genetic Algorithm	32
4.2	One-Point Crossover	34
4.3	Two-Point Crossover	34
4.4	Cycle Crossover	35
4.5	Mutation of a String	35
5.1	Flowchart of GO-LAO Algorithm	49
5.2	Division of the Global Design Space of the Camelback Function . . .	50
5.3	Surface Plot of Exact Camelback Function	52
5.4	Kriging Approximation of Camelback Function with Minimum (24) Sample Points	52
5.5	Kriging Approximation of Camelback Function with 80 Sample Points	53
5.6	Kriging Approximation of Camelback Function with 104 Sample Points	53
5.7	Kriging Approximation of Camelback Function with 120 Sample Points	54

5.8	Improvement in Accuracy with Additional Sample Points	54
5.9	Contour Plot of the Camelback function	57
5.10	Convergence Histories in GO-LAO for Camelback function	64
5.11	Convergence History in GA for Camelback function	65
5.12	Computational Time for the Camelback function	66
5.13	Convergence History in GO-LAO for AUV Problem	73
5.14	Convergence History in GA for AUV Problem	73
6.1	Design space of the Barnes problem	83
6.2	Exact contours of the Barnes problem	84
6.3	Contours of CRSA for the Barnes problem	85
6.4	Contours of error percentage for the Barnes problem	86
6.5	Post-optimality performed on the CRSA of the Barnes problem	86
6.6	Percentage Change in Objective Function with respect to Initial Optimal Design for the HPLC ten-bar problem	89
6.7	Percentage Error with respect to Exact Function for the HPLC ten-bar problem	90
6.8	Percentage Change in Objective Function with respect to Initial Optimal Design for the Control-Augmented Structure problem	91
6.9	Percentage Error with respect to Exact Function for the Control-Augmented Structure problem	93
A.1	Schematic for the Compound Valve Head	102
A.2	Visualization of a Simulation in which the AUV is Closing in on the Targeted Torpedo	109
A.3	AUV Mission Scenario	111
A.4	Schematic of the HPLC Structure	114
A.5	Controls Augmented Structure	116
A.6	Coupled System of the Controls-Augmented Structure problem	117

TABLES

5.1	EXACT GLOBAL OPTIMUM SOLUTIONS OF THE CAMELBACK FUNCTION	58
5.2	DIFFERENT GA TRIAL RUNS FOR CAMELBACK FUNCTION	60
5.3	GLOBAL OPTIMIZATION RESULTS FOR THE CAMELBACK FUNCTION	62
5.4	GA TRIALS FOR THE ENGINE DESIGN PROBLEM	68
5.5	GLOBAL OPTIMIZATION RESULTS FOR THE ENGINE DESIGN PROBLEM	70
5.6	GLOBAL OPTIMIZATION RESULTS FOR THE AUV PROBLEM	72
6.1	POST-OPTIMALITY RESULTS FOR THE BARNES PROBLEM	87
A.1	NOMENCLATURE FOR ENGINE MODEL	103
A.2	ENGINE DESIGN SPECIFICATION PARAMETERS	106
A.3	COEFFICIENTS FOR THE BARNES PROBLEM	113

SYMBOLS

\mathbf{x}	Vector of design variables
$Y(\mathbf{x})$	Undetermined functional form of the system
$F(\mathbf{x})$	A known regression model
$Z(\mathbf{x})$	A stochastic process
Cov	Covariance between two design points
\mathbf{R}	Coorelation matrix
R	Correlation function between two points
n_{dv}	number of design variable
n_s	number of sample points
θ_k	Parameter in a correlation function
p_k	Parameter in a correlation function
$\hat{\beta}$	Estimated coefficient in Kriging model
\mathbf{F}	Column vector with elements equal to 1.0
$\mathbf{r}^T(\mathbf{x})$	Correlation vector between \mathbf{x} and sample points
$\hat{Y}(\mathbf{x})$	Predicted estimate at untried point \mathbf{x}
$\hat{\sigma}^2$	Estimate of variance of Kriging model
e_i	Prediction error at i th omitted design site
Y_i	Exact function value at i th sample point
\check{Y}_i	Predicted function value at i th omitted design site
$PRESS$	Prediction error sum of squares
SS_T	Total sum of squares
$R_{prediction}^2$	Measure of accuracy
$CRSA$	Cumulative response surface approximation
NL	Number of local regions
$f(\mathbf{x})$	Objective function
$\tilde{f}_i(\mathbf{x})$	RSA of the function $f(\mathbf{x})$ in i th local region
$\phi_i(\mathbf{x})$	i^{th} blending function
$\hat{F}(\mathbf{x})$	CRSA of the function $f(\mathbf{x})$
$d_i(\mathbf{x})$	Distance between point \mathbf{x} and i th sample point
s_i	Influence parameter of the i th local region
\mathbf{p}	Vector of system parameters
h_j	j^{th} equality constraint
g_i	i^{th} inequality constraint
\mathbf{x}^l	Lower bound on the design variable \mathbf{x}
\mathbf{x}^u	Upper bound on the design variable \mathbf{x}

L	Lagrangian function
λ_j	Lagrange Multiplier for j^{th} equality constraint
μ_i	Lagrange Multiplier for i^{th} inequality constraint
∇f	Gradient of objective function
H	Hessian of the response surface approximation
NI	Number of design iterates
$NLPOA$	Non-linear post-optimality analysis
$y(\mathbf{x})$	Part of the constraint dependent on design variable
$z(\mathbf{p})$	Part of the constraint dependent on system parameter
$g(\mathbf{x}, \mathbf{p})$	Constraint dependent on design variable and system parameter

ACKNOWLEDGMENTS

Before we get into the details of this thesis, I would like to take this opportunity to express my sincere gratitude to all the people who were a part of this research and gave me continuous support right from the beginning of the work.

I would like to thank my advisor Dr. John E. Renaud for his guidance and advice on various issues from time to time. His valuable guidance, continuous support and immense help at all the stages of this research enriched me with an insight into this field. His lucid understanding of the field provided me with a crystal perception into the various aspects of design and optimization. He enthusiastically contributed in driving my endeavors towards the achievement of desired objectives. Also, I would like to acknowledge Drs. Stephen Batill and Mihir Sen, the members of my examination committee, for their time and input.

I would like to thank all my colleagues and friends at Notre Dame, who were directly or indirectly involved at any stage of my research work. I would like to thank Victor Perez, Harish Agarwal, Weiyu Liu, Shawn Gano, Dhanesh Padmanabhan, Alejandro Espinoza, Andres Tovar and Neal Patel for their enlightening discussions and suggestions. Also I would like to thank the Department of Aerospace and Mechanical Engineering for the computational facilities. Finally, I would like to acknowledge the support of NSF and Office of Naval Research for funding this work.

CHAPTER 1

INTRODUCTION

In today's simulation based design environment, the design of large scale engineering systems, such as automobiles, aircraft or underwater vehicles often involves years of effort. Optimum designs are preferred in this cost-conscious world. In order to obtain the optimized designs, some optimization of the system should be performed. However, optimization of the multidisciplinary systems would take a long time to accomplish. Multidisciplinary Design Optimization (MDO) is usually a cost-intensive procedure due to the large number of high fidelity function calls. Optimization costs can be reduced by approximating the high fidelity model of the system by low fidelity approximation.

1.1 Response Surface Approximation

Response Surface Approximations (RSAs) are often used to approximate the response of the multidisciplinary systems. RSA are also referred to as *metamodels* or *surrogate* models by some researchers [18, 54]. These RSAs are low fidelity representations of the actual model of the system. Response surface approximations are generated by [21]:

1. Designing a set of experiments that will yield adequate and reliable measurements of the response of the system.
2. Determining a mathematical model that best fits the data collected from the design chosen above, by conducting appropriate tests of hypotheses concerning the model's parameters.

The simplest representation of a RSA is a quadratic approximation. More complex and accurate representations include Multivariate Adaptive Regression Splines, Neural Networks, Radial Basis Functions, Kriging and others.

1.2 Classification of Approximation

Approximations can be constructed in two domains: local and global. The simplest difference between local and global approximation is the size of region where the approximation is valid. However, it is not quite appropriate to distinguish local and global approximations just by the size of the region. The actual function could be very simple in a large region, and very complex and multi-modal in a small region. So, the type of approximation, local or global, may also depend on how complicated the actual function is. In this research, local approximations are constructed to provide an accurate representation of the model.

1.3 Optimization Categories

Optimization can be classified into two categories: local and global. The basic difference between local and global optimization is the size of region where optimization conditions hold. A local optimum has an extreme function value as compared to

the points contained in the small neighborhood. However, the global optimum has the extreme function value amongst all the points in the whole design space. There are different local optimization methods which converge to a local optimum. Some of them are *pattern search*, *conjugate direction*, *generalized reduced gradient* and many others. Some of the popular global optimization methods are *genetic algorithm* and *simulated annealing*, which make an attempt to locate the global optimum. However none of these global methods guarantee the location of global optimum. In principle, it is only possible to prove the global convergence by exhaustive search.

1.4 Research Problem Overview

While designing complex engineering systems, the designer always feels the need to find the global optimum design. There are numerous well-established methods for local optimization. There are global optimization methods as well which make an attempt to identify the global optimum design. The famous ones are genetic algorithm, simulated annealing and variations of them *i.e.* hybrid algorithms. The designs can become obsolete in a short time span in this fast paced world. Thus the designer is always concerned with the efficiency of the algorithm. This research is driven by the need for an efficient global optimization method. So the focus of this research is to develop an efficient global optimization strategy. This investigation is a proof of concept development effort. Refinement recommendations are included as part of future research recommendations. Another aspect of this research is the development of a non-linear post optimality analysis strategy.

1.4.1 Global Optimization

A strategy for global optimization has been developed by using local approximate optimization. The task of developing a global optimization strategy is approached in the following manner:

- **Division into local regions:** The basic paradigm in this research is to divide the global domain into local domains. The global design space is divided into a number of local regions. The number of regions grows with the dimensionality of the problem. The number of local regions is manageable in small variable problems. However it is difficult to handle a large number of regions in bigger problems. A heuristic division scheme has been proposed to address this problem.
- **Constructing accurate approximation:** Usually RSAs are employed to generate an approximation of the computationally intensive functions. In this study, the Kriging models have been used to approximate the response of the system. Kriging approximations are constructed in each of the local regions. A minimum number of sample points; as decided by the dimensionality of the problem, are used to construct the Kriging model. The model quality is assessed and validated in terms of an inexpensive cross-validation measure of accuracy. Information from additional sample points is used to improve the quality of the approximation. The validated model is then used to predict the function value at a given design point.

- **Performing optimization:** Gradient based local optimization is performed in each of the local regions. To encounter the possibility of multiple optima in a particular local region, the strategy of multi-starting points is applied. The optima obtained from different local regions are further analyzed to provide an estimate of the global optimum solution.

1.4.2 Non-Linear Post-Optimality Strategy

Linear post-optimality analysis is based on linear estimates and thus does not provide an accurate estimate of the post-optimal design. Linear analysis is of limited applicability for non-linear MDO problems. This calls for the development of a non-linear strategy; which happens to be another contribution of this research. A non-linear post-optimality strategy has been formulated by addressing the following issues:

- **Constructing cumulative approximation:** Local RSAs are able to capture the non-linear response of the complex problems. The true nature of the function is extracted by the local approximations constructed in the local domain. These local RSAs are then blended to give a cumulative global response surface approximation (CRSA). Thus the non-linear behavior of the system is represented by a cumulative approximation.
- **Post-optimization over the CRSA:** The cumulative approximation is constructed for the objective function and all the constraints. A new optimization problem is formulated as the minimization of CRSA of the objective function subject to the CRSA of constraints. The post-optimal designs are obtained

by performing optimization over the CRSA formulation. Thus an accurate estimate of the post-optimal solution can be obtained at a reduced cost, as inexpensive approximations are used.

This non-linear strategy provides more accurate estimates of the post-optimal solution than the linear method.

1.5 Overview of the Thesis

To address the research issues mentioned above, this thesis is organized as follows. Chapter 2 provides a brief description of Kriging models. It presents the mathematical model on which the Kriging approximation is based. Some noteworthy characteristics of Kriging models are discussed in the chapter. A cross-validation measure of accuracy is used for evaluating the fidelity of the Kriging approximations. A method for improving the accuracy of the approximation is discussed.

The local RSAs are blended to construct a cumulative approximation via the blending function, discussed in Chapter 3. This chapter deals with issues regarding the generation of a cumulative approximation. The properties of the blending function are also discussed. Methods for improving the accuracy of the cumulative approximation are described.

Chapter 4 describes one of the global optimization methods, genetic algorithms in detail. A brief description of different steps involved in the genetic algorithm are provided. Variants of these steps are also mentioned.

The two original contributions of this research are described in detail in Chapters 5 and 6. The primary contribution of this research is the development of a

new global optimization algorithm, which is discussed in Chapter 5. The details of different steps of the algorithm are provided. The strategy for obtaining global optimum solutions is formulated. Different steps of the algorithm have been illustrated with suitable examples. The strategy is tested on several test problems. A comparison between the proposed method and genetic algorithms is made.

The non-linear post-optimality strategy is presented in Chapter 6. The underlying principle of the method is provided along with a comparison to a linear strategy. This chapter provides the mathematical formulation of the non-linear strategy. Advantages of the non-linear strategy have been pointed out. The strategy has been compared with linear post-optimality analysis for several test problems. The findings and summary of this aspect of the research are discussed.

A brief summary and conclusion for this research, accompanied by some recommendations for future work are presented in Chapter 7.

CHAPTER 2

KRIGING APPROXIMATION

2.1 Kriging Models

Kriging Models have their origins in mining and geostatistical applications which deals with spatially and intercorrelated data [28]. Many computer analysis codes are deterministic and therefore are not subjected to measurement error. Based on this observation, some statisticians developed *design and analysis of computer experiments* (DACE) for deterministic computer-generated data based on the Kriging method [48, 49]. The Kriging method used in this study is described in detail in the following sections.

2.2 Mathematical Formulation

The Kriging model is expressed as a combination of a regression model and a stochastic process:

$$Y(\mathbf{x}) = F(\mathbf{x}) + Z(\mathbf{x}) \quad (2.1)$$

where $Y(\mathbf{x})$ is the unknown function of interest, $F(\mathbf{x})$ is a known regression model and $Z(\mathbf{x})$ is a stochastic process with mean zero and variance σ^2 . While $F(\mathbf{x})$ globally approximates the design space, $Z(\mathbf{x})$ creates localized deviations so that

the Kriging model interpolates the n_s sampled data points. The covariance matrix of $Z(\mathbf{x})$ is given by:

$$Cov[Z(\mathbf{x}^i)Z(\mathbf{x}^j)] = \sigma^2 \mathbf{R}[R(\mathbf{x}^i, \mathbf{x}^j)] \quad (2.2)$$

where \mathbf{R} is the correlation matrix and $R(\mathbf{x}^i, \mathbf{x}^j)$ is the correlation function between any two of the n_s sampled data points \mathbf{x}^i and \mathbf{x}^j . \mathbf{R} is a $(n_s \times n_s)$ symmetric matrix with diagonal elements equal to 1.0. The correlation function $R(\mathbf{x}^i, \mathbf{x}^j)$ is specified by the user [48, 49]. A variety of correlation functions can be selected from Simpson *et al.* [53]. The general correlation function is of the form:

$$R(\mathbf{x}^i, \mathbf{x}^j) = \exp\left[-\sum_{k=1}^{n_{dv}} \theta_k |x_k^i - x_k^j|^{p_k}\right] \quad (2.3)$$

where $\theta_k \geq 0$ and $0 < p_k \leq 2$. Here n_{dv} is the number of design variables, θ_k and p_k are unknown correlation parameters used to fit the model. The parameter p_k can be interpreted as an indicator of increasing the smoothness of the response surface. The θ 's indicate how local the estimate is. If θ_k is large, only data points in the immediate vicinity of a given point are correlated with that point. If θ_k is small, points further away are highly correlated and have influence on that point. Sometimes using a single correlation parameter gives good results [49]; however a different θ for each design variable is used in the present work.

The predicted estimates, $\hat{Y}(\mathbf{x})$ of the response $Y(\mathbf{x})$ at the untried values of \mathbf{x} are given by:

$$\hat{Y} = \hat{\beta} + \mathbf{r}^T(\mathbf{x})\mathbf{R}^{-1}(\mathbf{Y} - \hat{\beta}\mathbf{F}) \quad (2.4)$$

where \mathbf{Y} is the column vector of length n_s which contains the response at each sample point and \mathbf{F} is a column vector of length n_s which has elements equal to

1.0, when $F(\mathbf{x})$ is taken as a constant. In Eq. (2.4), $\mathbf{r}^T(\mathbf{x})$ is the correlation vector of length n_s between an untried \mathbf{x} and the sampled data points $[\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^{n_s}]$. In Eq. (2.4), $\hat{\beta}$ is estimated using least squares regression as shown below:

$$\hat{\beta} = (\mathbf{F}^T \mathbf{R}^{-1} \mathbf{F})^{-1} (\mathbf{F}^T \mathbf{R}^{-1} \mathbf{Y}). \quad (2.5)$$

The estimate of the variance is given by:

$$\hat{\sigma}^2 = \frac{(\mathbf{Y} - \hat{\beta} \mathbf{F})^T \mathbf{R}^{-1} (\mathbf{Y} - \hat{\beta} \mathbf{F})}{n_s}. \quad (2.6)$$

The maximum likelihood estimates (MLE), for θ_k and p_k are found from:

$$\text{Minimize} \quad f(\theta_k) = -\frac{(n_s \ln(\hat{\sigma}^2) + \ln |\mathbf{R}|)}{2} \quad (2.7)$$

$$\text{Subject to:} \quad \theta_k \geq 0$$

$$0 < p_k \leq 2$$

where both $\hat{\sigma}^2$ and $|\mathbf{R}|$ are functions of θ_k and p_k . This forms a n_s dimensional unconstrained non-linear sub-optimization problem which needs to be solved to fit the Kriging model. The reader is referred to Lophaven *et al.* [24, 25] for additional implementation aspects of Kriging models.

2.3 Illustration of Kriging Approximation

The surface plot of the Camelback function in Appendix A.1 is shown in Fig. 2.1. As can be seen, the function is highly non-linear and multimodal in nature. The Kriging approximation of this function is constructed in Fig. 2.2. This figure gives the reader a visual feel for Kriging's ability to approximate the response of the non-linear function. It is clearly evident from Fig. 2.2 that Kriging models provide

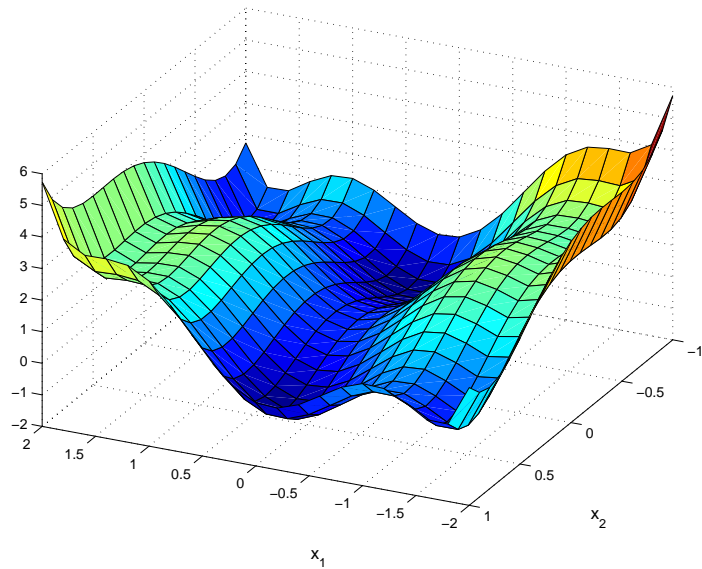


Figure 2.1. Surface Plot of the Exact Camelback Function

a good approximation of a complex function. Thus the ability to accurately map non-linear complex functions by the use of Kriging models supports the decision to use them in the present work.

2.4 Features of Kriging Models

Kriging models are gaining popularity for approximating the mapping process of a deterministic computer model. Certain important features of Kriging models are listed here:

- Kriging models offer a lot of flexibility to capture the response of the system. The flexibility of Kriging models is due to the fact that there are a large number of correlation functions to choose from [53]. The model parameters control the model's properties, such as smoothness of the response surface,

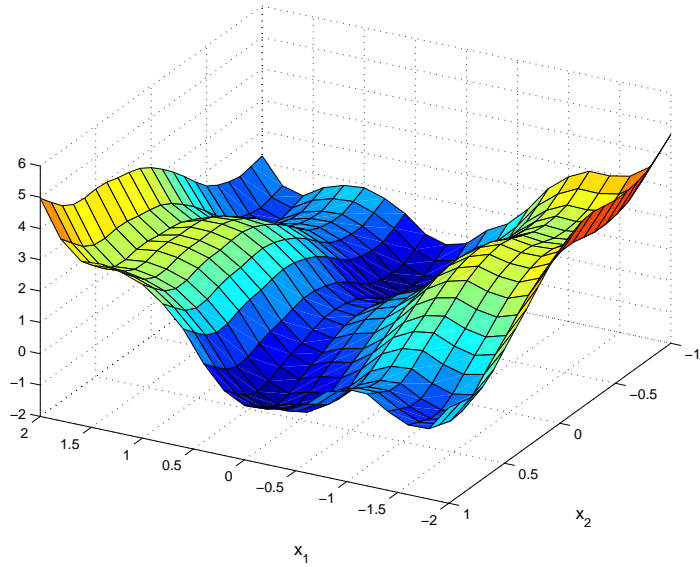


Figure 2.2. Surface Plot of the Kriging Approximation of the Camelback Function

differentiability and the ability to vary the influence of each design variable. These model parameters are capable of approximating the response of most functions ranging from simple to multimodal non-linear functions. Some researchers have compared various metamodeling techniques and concluded that Kriging models approximate the response of the system to a better extent than other RSAs [18, 54].

- Kriging models have a property, “convergence of extrapolation”, which is another important feature that makes the method more attractive than many other approximation techniques. An extrapolation of the Kriging approximation of $y = \sin(x)$ in a wide region of interest is illustrated in Fig. 2.3. It shows that a Kriging approximation converges to the average value of sample points, instead of approaching infinity as several other approximation techniques do.

This feature is very useful in the area of global approximation, where enlarging the region of interest is often encountered.

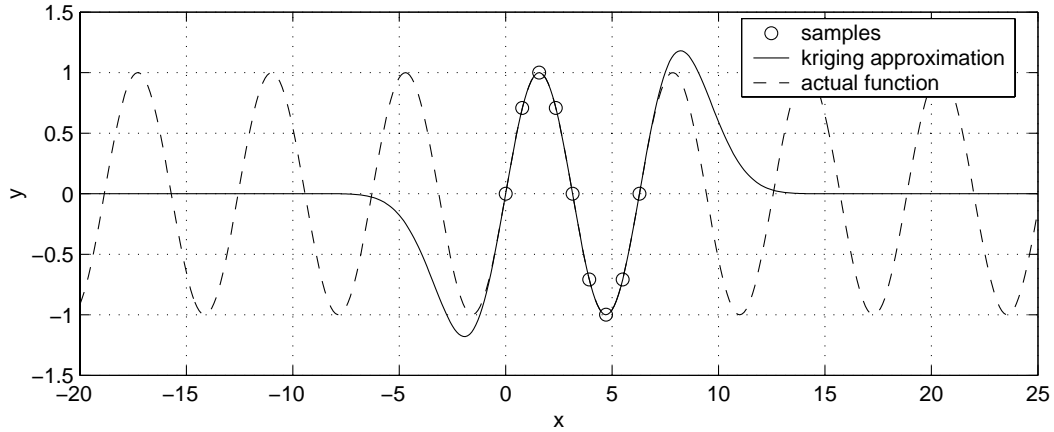


Figure 2.3. Extrapolation of a Kriging approximation of $y = \sin(x)$ in a wide region

- Kriging models are an interpolating approximation and thus exactly pass through the sample points. It is not a regression technique where the approximation provides a best fit to the sample points. Most of the high fidelity simulation tools like Finite Element Analysis packages and Computational Fluid Dynamics packages are deterministic codes. Kriging approximations are good choice for deterministic computer models since it interpolates the observed or known sample points [54].

2.5 Disadvantages of Using Kriging Models

The major disadvantage of the Kriging process is that model construction can be very time-consuming. Determining the maximum likelihood estimates of the θ and p parameters used to fit the model is a n_s dimensional optimization problem, which

can require significant computational time if the sample data set is large. However, with the advent of high performance computing, the evaluation of maximum likelihood estimates is still possible in a short time frame as compared to performing a function evaluation using the high fidelity simulation codes. If the underlying function is highly non-linear in a moderate number of factors (less than 50, say), then Kriging may be the best choice despite the added complexity [54]. Thus despite the complexity and time-consuming model construction process, Kriging models are widely used owing to their prediction accuracy.

The correlation matrix can become singular if multiple sample points are spaced close to one another or if the sample points are generated from specific design of experiments. This problem of singularity can be avoided by choosing points which are not too close to each other. The ill-conditioned correlation matrix is handled by using a Cholesky factorization [25]. Fitting problems have been observed with some full factorial designs and central composite designs using Kriging models [30].

2.6 Model Validation

Once the model is constructed, there should be some measure of accuracy which would validate the acceptance of the model. The metamodel or response surface quality is assessed in terms of two aspects.

- The first evaluates the capability of the model in reproducing the observed data points. This measure of accuracy is relevant only in the case of regression models. However, since the Kriging models are interpolating, this measure of accuracy is meaningless.

- The capability of the metamodel to approximate the exact function is the second measure of accuracy. Some measures of accuracy like Root Mean Square error (RMSE), Relative Average Absolute error (RAAE) and others are described in references [18, 27].

2.6.1 Measures of Error

One of the popular measures of accuracy is the Root Mean Square Error (RMSE). A large number of additional sample points are required to be evaluated to compute the RMSE. These sample points are different than the ones used in model construction. However this could be computationally intensive for the high fidelity simulation codes used in the design engineering. Thus an alternative solution *cross-validation* is used to save the time in model-validation. A popular cross-validation method is a leave- k -out strategy, in which, k points are left out of the constructed metamodel. The resulting error is then calculated based on the omitted points. Mitchell and Morris [33] and Osio and Amon [37] provide a description of this technique for $k = 1$, leave-*one*-out strategy. Meckesheimer *et al.* [31] investigated computationally inexpensive assessment method based on leave- k -out cross validation and developed guidelines for selecting k for different types of metamodels.

2.6.2 Leave-*one*-out Cross-Validation Strategy

In the leave-*one*-out strategy, one sample point is left out and the remaining $n_s - 1$ points are used to fit the Kriging model. This Kriging model is used to

predict the function value at the omitted site and is denoted by \check{Y}_i . The prediction error for the i th left out point is:

$$e_i = Y_i - \check{Y}_i \quad (2.8)$$

where Y_i is the exact function value at the omitted site. This process is repeated by omitting each sample point, one at a time. These n_s errors of prediction are used to calculate Prediction Error Sum of Squares (*PRESS*):

$$PRESS = \sum_{i=1}^{n_s} e_i^2 = \sum_{i=1}^{n_s} (Y_i - \check{Y}_i)^2 \quad (2.9)$$

The above discussed method for calculating *PRESS* statistic would also be computationally expensive, where an optimization is required to solve each new MLE.

2.6.3 Approximate Cross Validation

Mitchell and Morris [33] suggested a method that is not as exhausting as the traditional method discussed above. The same correlation parameters (θ_k, p_k) are used when creating the n_s Kriging models using the remaining $n_s - 1$ points. Recall that θ_k, p_k are the correlation parameters used to fit the Kriging model and n_s is the number of sample points from which the model is constructed. The error of prediction at the deleted site is:

$$e_i = q_i(g_i - \hat{\beta}w_i) \quad (2.10)$$

where $\hat{\beta}$ is given in Eq. 2.5 and

$$q_i = \frac{1}{\mathbf{R}_{i,i}^{-1}} \quad (2.11)$$

$$g = \mathbf{R}^{-1}\mathbf{Y} \quad (2.12)$$

$$w = \mathbf{R}^{-1}\mathbf{F} \quad (2.13)$$

where $\mathbf{R}, \mathbf{F}, \mathbf{Y}$ are described in sec. 2.2. This method is significantly faster since it requires evaluating and inverting the correlation matrix, \mathbf{R} , only once instead of doing so for each subset of points. The main difference between this method and traditional method is that the parameter $\hat{\beta}$ is not updated for each subset of the omitted points. The total sum of squares is given by:

$$SS_T = \mathbf{Y}^T\mathbf{Y} - \frac{\sum_{i=1}^{n_s} Y_i^2}{n_s} \quad (2.14)$$

where n_s is the number of sample points, Y_i is the exact function value at i th sample point and \mathbf{Y} is the vector containing the exact function value at the sample points.

The *PRESS* statistic and total sum of the squares (SS_T) are used to compute the following measure of accuracy [26]:

$$R_{prediction}^2 = 1 - \frac{PRESS}{SS_T} \quad (2.15)$$

This statistic provides validation of the prediction made by the Kriging model. If $R_{prediction}^2$ lies close to 1.0 then the prediction made by Kriging model can be accepted with a good amount of confidence. If the value deviates to a large extent from 1.0 then it is a bad approximation and the designer needs to re-build the model by taking additional sample points. This approximate cross-validation measure of accuracy is used in this work to assess the quality of Kriging models.

2.7 Summary

This chapter provides a brief description of Kriging models. It presents the mathematical model on which the Kriging approximation is based. The effect of model parameters in controlling the shape and smoothness of the response surface has been discussed. Some merits and demerits of using Kriging approximations have been described along with implementation details. A basis for validating the model is presented, which gives a measure of accuracy of the prediction made by the Kriging model. Thus, different aspects of model construction along with model validation have been provided.

CHAPTER 3

CUMULATIVE APPROXIMATION

Accurate response surface approximations are easily constructed in a small localized region. In one component of this research, accurate representations of local regions by RSAs are used to map an approximation of the global domain. Local response surface approximations can be blended together to provide a cumulative approximation. The *cumulative response surface approximation* will be referred to as CRSA. This CRSA is used as a basis for the non-linear post-optimality technique developed in this research.

3.1 Spline Interpolation for Cumulative Approximation

The cumulative approximation is based on the idea of interpolation between the function values at different points. This problem is well-known from the field of computational geometry and computer-aided design. In that case, interpolation is used to construct smooth curves and surfaces governed by a set of control points. Mortenson [34] and Farin [11] provide a detailed description of the interpolating schemes used in computational geometry.

The different types of spline curves and surfaces share the common feature that they are constructed from a set of blending functions related to the control points [8]. The blending function of a given point is a measure of influence of that point on the overall shape of the curve. The blending function attains a maximum near its control point and decreases in value for far away points. Chandila [5] provides a complete description of the behavior of the Rational B-Spline blending function's influence on the resulting approximation of the true response.

3.2 Drawbacks of Spline Interpolation

The main drawback of Spline interpolations is their requirement for control points to be arranged in a regular pattern. Thus, interpolation in a given dimension requires a minimum number of control points, for instance two for curves, four for surfaces, eight for volumes and so on. This is an undesirable feature of the spline interpolation which puts a constraint on the number or position of control points. The requirement for a regular pattern is related to the fact that the spline interpolation is performed with separate blending functions in each coordinate direction. Instead of using the above strategy, Rasmussen [44] suggested a different approach for constructing cumulative approximations. In this method the values of the blending functions depend on the distance from the control points.

3.3 Cumulative Response Surface Approximation (CRSA)

Rasmussen [44] provided a scheme for cumulative approximation. His method is based on the idea of improving the approximation of the function by including the

information from the previous design points visited during the sequential approximate optimization. He used an exponential blending function to generate cumulative approximations, which he refer to as CA. By using the simple exponential blending function which decays with distance, the function values at two different points can be interpolated. The different local response surface approximations are blended via the blending function suggested by Rasmussen. This is used to generate cumulative response surface approximation (CRSA), which is essentially the same as CA; but a different nomenclature.

3.4 Mathematical Formulation of CRSA

Let $f(\mathbf{x})$ be the objective function and NL be the number of local regions in which a local approximation is constructed. The local response surface approximations $\tilde{f}_i(\mathbf{x})$ are archived for each i th local region. The cumulative response surface approximation (CRSA) of the function $f(\mathbf{x})$ is given by [44]:

$$\hat{F}(\mathbf{x}) = \frac{\sum_{i=1}^{NL} \phi_i(\mathbf{x}) \tilde{f}_i(\mathbf{x})}{\sum_{i=1}^{NL} \phi_i(\mathbf{x})} \quad (3.1)$$

where $\phi_i(\mathbf{x})$ is the blending function corresponding to point \mathbf{x}_i . The blending function is defined as:

$$\phi_i(\mathbf{x}) = \exp \left[\frac{-d_i(\mathbf{x})}{s_i^2} \right] \quad (3.2)$$

where s_i is the influence parameter and $d_i(\mathbf{x})$ is the square of the distance from point \mathbf{x}_i . The blending function ϕ_i has a maximum value of 1 at $\mathbf{x} = \mathbf{x}_i$ and minimum of 0 for large distances $d_i(\mathbf{x}) \rightarrow \infty$. The denominator of Eq. 3.1 is the sum of the blending functions and has its primary effect in rendering the sum of influences to

unity in regions where CRSA is influenced by several local regions. Fig. 3.1 shows the behavior of the blending function ϕ_i as the distance d_i increases. The distance

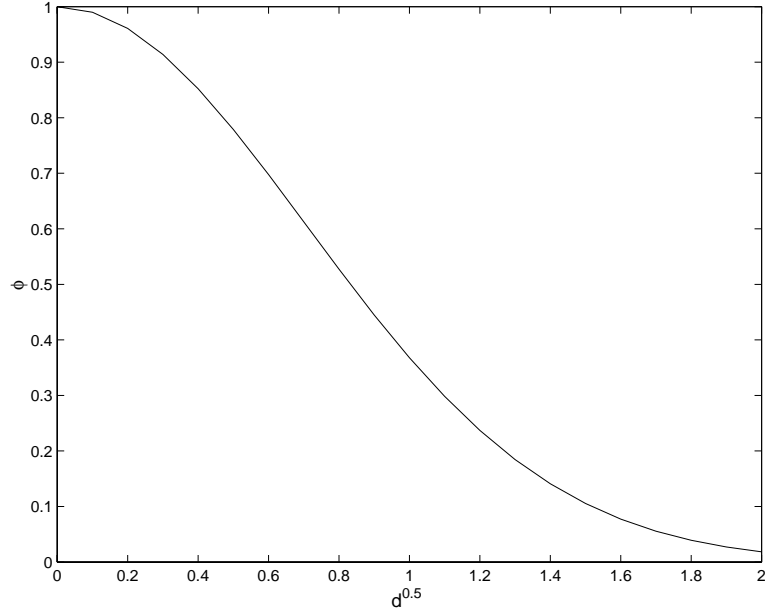


Figure 3.1. Effect of Distance on the Blending Function

d_i is defined as:

$$d_i(\mathbf{x}) = \sum_{k=1}^{n_{dv}} (x^k - x_i^k)^2 \quad (3.3)$$

where n_{dv} is the number of design variables. The influence parameter s_i is a measure of the influence region of each blending function ϕ_i . Note that, Rasmussen [44] uses a constant s , called normalization factor, for all the regions. However in the present study, each local region can have a different region of influence. A large value of s_i will give neighboring regions a relatively large influence on each other. This creates a smooth transition between the two neighboring regions, but gives less accurate local approximations in each individual region. Smaller values of s_i will create more abrupt transitions and more accurate local approximations. Thus the value of the

influence parameter should be chosen carefully. In this work, the influence region s_i has been taken to be equal to the size of the i th local region.

3.5 Illustration of CRSA

The cumulative response surface approximation is illustrated in this section by constructing CRSA's of two test functions. Implementation issues related to constructing a good CRSA are also discussed.

3.5.1 CRSA of Simple Test Function

The CRSA of a test function $f(x) = x^2 + x^3 + \sin(x)$ is shown in Fig. 3.2. The sample points were taken at $x = 1, 2, 3, 4, 5$. The size of the local region around each sample point was taken to be 0.5, which in turn is the influence region for each local region. As you can notice from Fig. 3.2, that a good cumulative approximation is constructed for the simple test function. The function value matches exactly at the sample points. However as you go away from the sample points, certain error starts to incur. Small error in the function value can be noticed at $x = 0$.

3.5.2 CRSA of Complex Test Function

Now, a more complex objective function is used to evaluate and obtain a good cumulative approximation. In this case, function $f(x) = 0.5x^2 - 0.1x^3 + \sin(2x)\exp(x)$ was used. The same set of sample points and influence region were used. Fig. 3.3 shows the CRSA of this complex function. Unfortunately this time, spikes are observed at the boundaries of local regions owing to the complexity of the function.

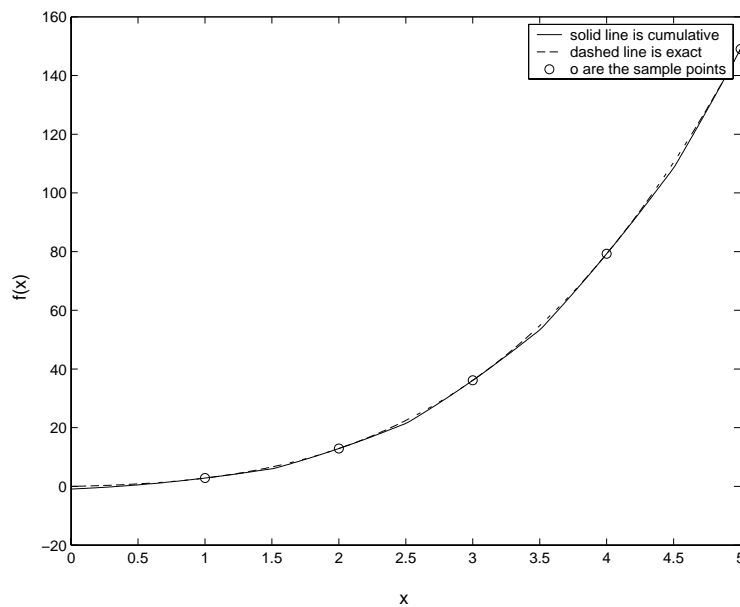


Figure 3.2. CRSA of the Simple Test Function

The spikes can be reduced and the function can be made smooth at the boundary of local regions in the following two ways.

- One way, as discussed earlier, is to vary the influence parameter s_i . Larger values of the influence parameter provides smooth transition between the boundaries. However, as mentioned in Sec. 3.4, large values of influence region will give less accurate approximation in each individual region. One may try to play and experiment with the parameter s_i to obtain smooth transition. However this would be a trial and error technique. Thus this method is not advisable in an environment where the quality of the approximation matters and a priori experimentation is not possible.

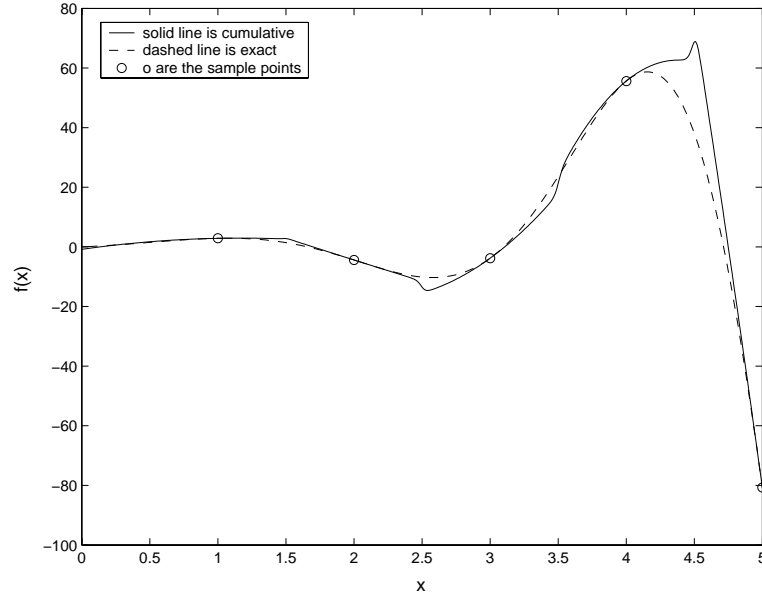


Figure 3.3. CRSA of the Complex Test Function

- The other option is to add some more sample points at the points of abrupt transition. Additional sample points improve the quality of the approximation. The criteria for location of additional sample points is based on the accuracy of the approximation. Sample points are added in the region where the surrogate model accuracy is unacceptable.

For the above function, if the number of sample points are increased, *i.e.* add another sample point at $x = 4.5$, the spikes in the cumulative approximation can be reduced. This is clearly shown in Fig. 3.4.

If the sample points are further increased, the quality of the approximation improves to a greater extent. Addition of sample point at $x = 2.5$ along with previously added point at $x = 4.5$, further reduces the spikes in the cumulative

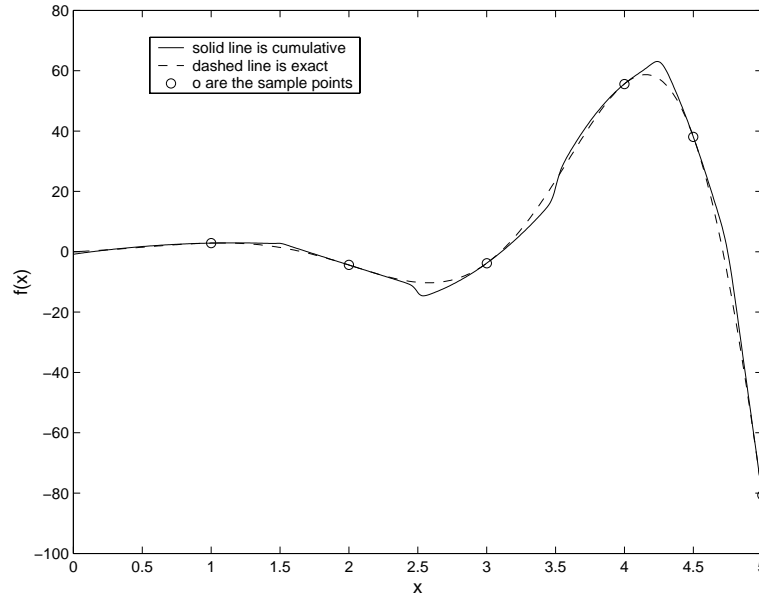


Figure 3.4. CRSA of the Complex Test Function With Additional Point

approximation. It is evident from Fig. 3.5, that additional sample points increase the accuracy of the cumulative approximation.

3.6 Summary

This chapter summarizes the method of constructing a cumulative response surface approximation from local approximations. Simple exponential blending functions are used to blend the local approximations. The properties of the blending function were described. Issues related to smooth or abrupt transition along the boundaries were also discussed. The accuracy of the cumulative approximation can be improved by using information at additional sample points.

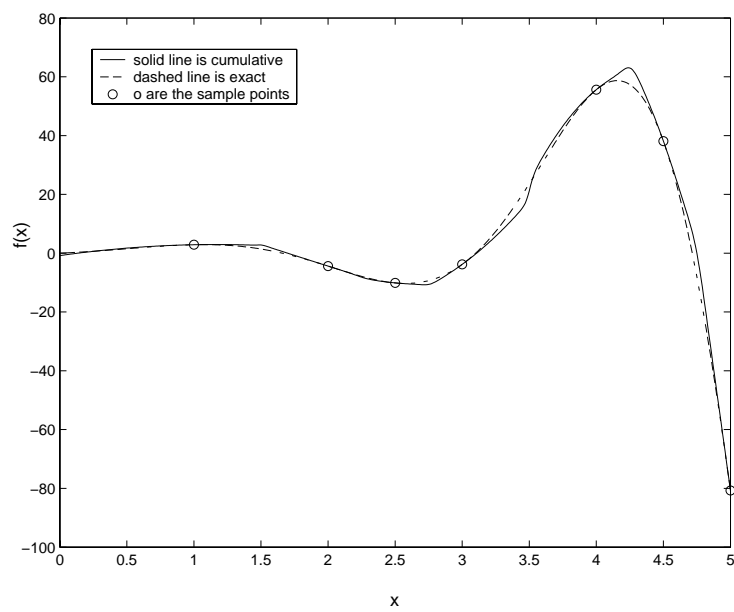


Figure 3.5. CRSA of the Complex Test Function With Another Additional Point

CHAPTER 4

GENETIC ALGORITHM

4.1 What are Genetic Algorithms?

Conventional search techniques, such as hill-climbing, are often incapable of optimizing non-linear multi-modal functions. In such cases, a random search method might be required. Genetic Algorithms are general purpose search and optimization methods applicable to a wide variety of real-life problems [63]. These algorithms are based on concepts which have a corollary in biological evolution. They operate on a population of potential solutions applying the principle of survival of the fittest. At each generation, a new population is created by the process of selecting individuals according to their level of fitness in the problem domain and breeding them together using operators borrowed from natural genetics. This process leads to the evolution of individuals which are better suited to the environment than the individuals that they were created from.

4.2 Overview of GAs

GAs do not use much knowledge about the problem to be optimized and do not deal directly with the parameters of the problem. They work with codes which represent the parameters. Thus, the first issue in a GA application is how to represent the problem parameters. They operate with a population of possible solutions, not just one possible solution. Hence the second issue becomes, how to create the initial population of possible solutions. Another issue in a GA application is how to select or devise a suitable set of genetic operators. Finally, as with other algorithms, GAs have to know the quality of already found solutions to improve them further. This calls upon evaluation of some fitness merit.

4.3 GA Application Issues

Implementation issues for GAs are described in this section [42]. There are variants of these procedures in the literature [16, 58].

4.3.1 Representation

The parameters to be optimized are usually represented in a string form since genetic operators are suitable for this type of representation. The method of representation has a major impact on the performance of the GAs. Different representations schemes might cause different performances in terms of accuracy and computational time.

There are two common representation methods for numerical optimization problems [9, 32]. The preferred method is the binary string representation method.

Various binary coding schemes can be found in the literature, for example uniform coding and gray scale coding. The second representation method is to use a vector of integers or real numbers, with each integer or real number representing a single parameter.

When a binary representation scheme is employed an important issue is to decide the number of bits used to encode the parameters to be optimized. Each parameter should be encoded with the optimal number of bits covering all possible solutions in the solution space. When too few or too many bits are used the performance can be adversely affected.

4.3.2 Creation of Initial Population

At the start of optimization, the GA requires a population of initial solutions. There are two ways of forming this initial population. The first consists of using randomly produced solutions created by a random number generator. This method is preferred for problems about which no a priori knowledge exists or for assessing the performance of an algorithm.

The second method employs a priori knowledge about the given optimization problem. Using this knowledge, a set of requirements is obtained and solutions which satisfy those requirements are collected to form an initial population. In this case, the GA starts the optimization with a set of approximately known solutions and therefore converges to an optimal solution in less time than the previous method.

4.3.3 Genetic Operators

The flowchart of a simple GA is given in Fig. 4.1. There are three common genetic operators: *selection*, *crossover* and *mutation*. An additional reproduction operator, *inversion*, is sometimes also applied. Some of these operators were inspired by nature. It is not necessary to employ all of these operators in a GA because each functions independently of the others. The choice or design of operators depends on the problem and the representation scheme employed. For instance, operators designed for binary strings cannot be directly used on strings coded with integers or real numbers.

Selection: The aim of the selection procedure is to reproduce more copies of individuals whose fitness values are higher than those whose fitness values are low. The selection procedure has a significant influence on driving the search towards a promising area and finding good solutions in a short time. However, the diversity of the population must be maintained to avoid premature convergence and to reach the global optimal solution. In GAs there are mainly two selection procedures: *proportional selection* and *ranking-based selection* [57].

Proportional Selection is usually called “roulette wheel” selection since its mechanism is reminiscent of the operation of a roulette wheel. Fitness values of individuals represent the widths of the slots on the wheel. After a random spinning of the wheel to select an individual for the next generation, individuals in slots with large widths representing high fitness values will have a higher chance to be selected.

One way to prevent rapid convergence is to limit the range of trials allocated to any single individual, so that no individual generates too many offspring. The

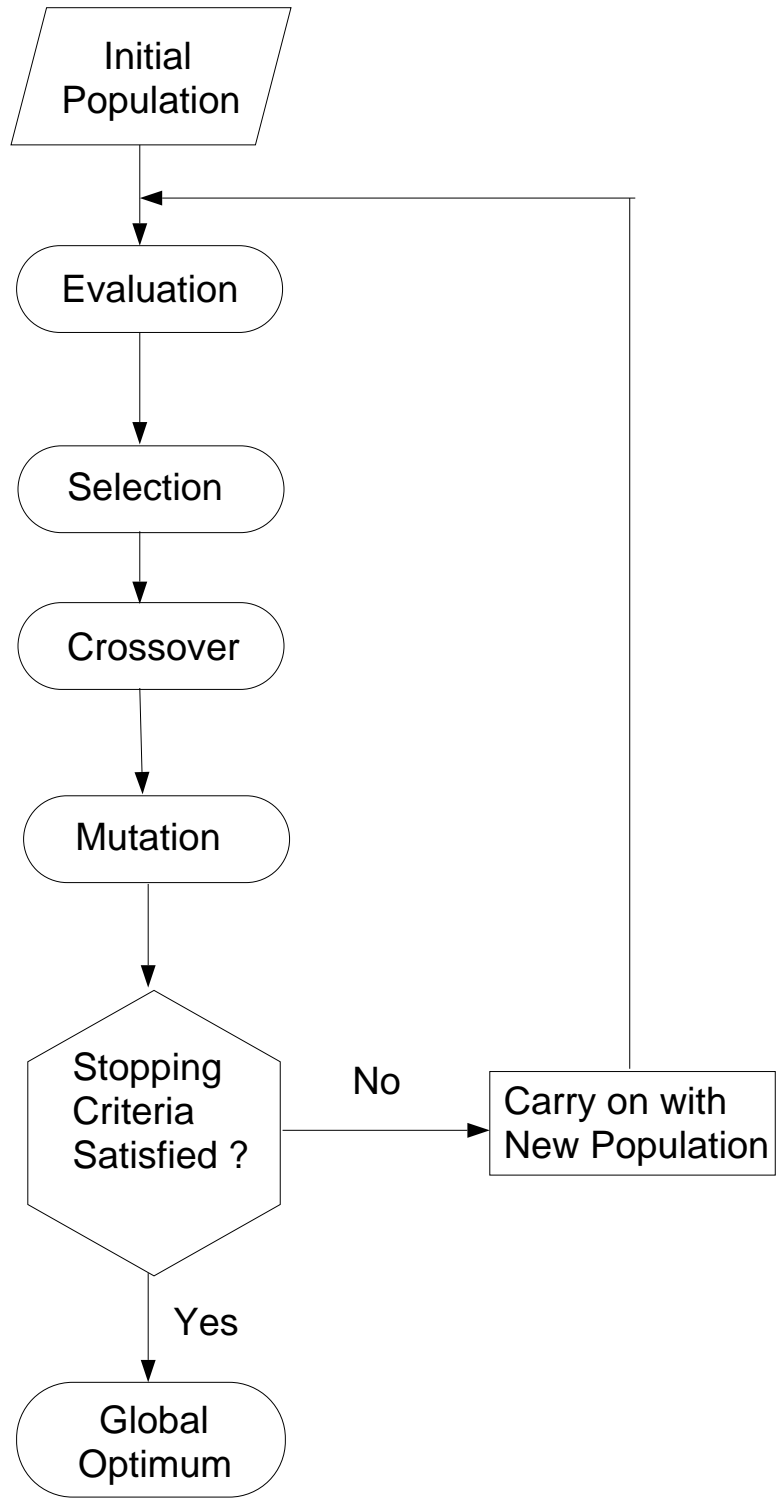


Figure 4.1. Flowchart of Genetic Algorithm

ranking-based procedure is based on this idea. According to this procedure, each individual generates an expected number of offspring which is based on the rank of its fitness value [1].

Crossover: This operation is considered the one that makes the GA different from other algorithms, such as dynamic programming. It is used to create two new individuals (children) from two existing individuals (parents) picked from the current population by the selection operation. There are several ways of doing this. Some common crossover operations are *one-point crossover*, *two-point crossover*, *cycle crossover* and *uniform crossover*.

One-point crossover is the simplest crossover operation. Two individuals are randomly selected as parents from the pool of individuals formed by the selection procedure and cut at a randomly chosen point. The tails, which are the parts after the cutting point, are swapped and two new individuals (children) are produced. Note that this operation does not change the values of the bits. Examples are shown in Figures 4.2-4.4 for different crossover operations.

Mutation: In this procedure, all individuals in the population are checked bit by bit and the bit values are randomly reversed according to a specified rate. Unlike crossover, this is a unary operation. That is, a child string is produced from a single parent string. The mutation operator forces the algorithm to search new areas. Eventually, it helps the GA avoid premature convergence and find the global optimal solution. An example is given in Figure 4.5.

Parent 1	1 0 0 0 1 0 0 1 1 1 1
Parent 2	0 1 1 0 1 1 0 0 0 1 1
Child 1	1 0 0 0 1 1 0 0 0 1 1
Child 2	0 1 1 0 1 0 0 1 1 1 1

Figure 4.2. One-Point Crossover

Parent 1	1 0 1 0 0 0 1 1 0 1 0
Parent 2	0 1 1 0 1 1 1 1 0 1 1
Child 1	1 0 1 0 1 1 1 1 0 1 0
Child 2	0 1 1 0 0 0 1 1 0 1 1

Figure 4.3. Two-Point Crossover

Parent 1	1 2 3 4 5 6 7 8
Parent 2	a b c d e f g h
Child 1	1 b 3 d e 6 g 8
Child 2	a 2 c 4 5 f 7 h

Figure 4.4. Cycle Crossover

Old String	1 1 0 0 0 1 0 1 1 1 0
New String	1 1 0 0 1 1 0 1 1 1 0

Figure 4.5. Mutation of a String

4.3.4 Fitness Evaluation Function

The fitness evaluation function acts as an interface between the GA and the optimization problem. The GA assesses the solutions for their quality according to the information produced by fitness function. In engineering design problems, functional requirements are specified to the designer who has to produce a design which performs the desired functions within pre-determined constraints. The quality of a proposed solution is usually calculated depending on how well the solution performs the desired functions and satisfies the given constraints. Fitness evaluation functions might be complex depending on the optimization problem at hand. Where a mathematical equation cannot be formulated for this task, a rule-based procedure can be constructed for use as a fitness function or in some cases both can be combined. Where some constraints are very important and cannot be violated, the solutions which do so can be eliminated in advance by appropriately designing the representation scheme. Alternatively, they can be given low probabilities by using special penalty functions.

4.4 Summary

The global optimization technique, genetic algorithms have been discussed in this chapter. Various underlying principles and implementation issues have been mentioned. These provide a basic understanding of genetic algorithms. The GAs use random operators to explore different regions. The genetic operators do not use gradient information from the problem to drive the algorithm. The random nature of genetic operators requires a large number of function evaluations. Genetic

algorithms try to identify possible global optimal solutions. However, there is no guarantee that the obtained solution is the true global optimum. In fact, it is difficult to prove the convergence in global optimization techniques, unless one resorts to an exhaustive search.

CHAPTER 5

GLOBAL OPTIMIZATION STRATEGY

5.1 Survey of Global Optimization Methods

Global optimization methods try to locate the solution which is optimum for the global design space. One of the most popular methods, *genetic algorithms* were discussed in detail in Chapter 4. Variants of genetic algorithms also exist in the literature [16]. Liu *et al.* [23] started with a simplex and then used simulated annealing to determine the direction and step of searching. Essentially, these hybrid methods use genetic algorithms or simulated annealing to find a good approximate point and then employ a local optimizer to find the global optima. The operators of the *differential evolution* algorithm are employed by Laskari *et al.* [22] to reduce the number of function evaluation in multi-start optimization methods. The starting points of their global optimization method are generated with dynamic search trajectories.

Xu [60, 61] has developed a hybrid global optimization algorithm which uses local optimizers and feasible point finders. The convergence of this hybrid method depends on the size of a bounding box used to locate the optima. A multi-start algorithm for global optimization using simplex elimination has been developed by

Oh and Lee [36]. They used multi-dimensional simplexes to express elimination regions. Their region elimination method consists of making a set of eliminated regions, checking adjacency between the current design point and the elimination region, and quitting local optimization for the eliminated design points. Their strategy consists of local optimization, region elimination and heuristic convergence check.

Schonlau [52] and Jones *et al.* [20] have developed an Efficient Global Optimization (EGO) method which uses a method of expected improvement for improving the metamodel accuracy. EGO adds the points which have either low objective function value or high uncertainty. Different infill sampling criterion have been explored by Sasena *et al.* [50] to improve the convergence in EGO. They have also discussed the influence of infill samples on the efficiency of the EGO algorithm. An overview of global optimization methods using response surfaces is provided by Jones [19].

5.2 Pitfalls of Genetic Algorithms

Genetic algorithms are a global optimization method based on the principle of “Survival of the Fittest”. They make an attempt to locate the global optimum solution. However, the ability to successfully identify the global optima is dependent on various factors. The factors influencing the convergence and a few disadvantages of genetic algorithms are described below:

1. These methods use a fitness function to determine the possible global optimum solution. The fitness of the members of a genetic population is evaluated by using the objective function only. GAs do not use any gradient information to capture the global optima.

2. The search is driven to different areas in the design space based on population modification via crossover or mutation genetic operations. These genetic operations are random in the sense that they randomly change the population to explore new areas. They do not use any problem information to explore new areas. For example, a randomly chosen bit is changed in the mutation process to search new areas. Although this might lead the search to unexplored areas, it will require unnecessary function evaluations. Obviously that is not advisable in a cost-conscious design environment.

3. Important control parameters of a simple GA include the population size (number of individuals in the population), crossover rate and mutation rate. The rate of convergence is dependent on these control parameters. Several researchers have studied the effect of these parameters on the performance of a GA [13, 51]. The main conclusions are as follows:
 - A large population size is recommended for identifying the global optimum solution. Since many samples from the search space are used, the probability of convergence to a global optimal solution is higher than when using a small population size. However a large population size means the simultaneous handling of many solutions and increases the computation time per iteration.
 - The crossover rate determines the frequency of the crossover operation. It is useful at the start of the optimization to discover a promising region. A low crossover frequency decreases the speed of convergence to such

an area. If the frequency is too high, it leads to saturation around one solution.

- The mutation operation is controlled by the mutation rate. A high mutation rate introduces high diversity in the population and might cause instability. On the other hand, it is usually very difficult for a GA to find a global optimal solution with too low of a mutation rate.

Therefore it is very important to choose suitable control parameters. Researchers select the values of the control parameters to suit the particular problems they are working with. It is very difficult to choose suitable control parameters for a problem a priori. One could try to use the recommended values of previous studies, which may or may not work for different problems.

5.3 Global Optimization Using Local Approximate Optimization (GO-LAO)

In this fast paced world, designs can become obsolete in a short span of time. Thus the designer should be able to provide new optimum designs in a short time frame. This research is propelled by the need for an efficient global optimization technique. A new global optimization method based on local surrogate models is developed in this research. In this approach, multi-start gradient based optimization over local approximations is performed to provide an estimate of the global optimum design. Thus the name *global optimization using local approximate optimization* (GO-LAO) follows from the underlying principle.

The basic assumption in this approach is that evaluating the system analysis is computationally intensive and incurs a large computational cost. Thus it is not

possible to use the exact high fidelity system analysis for optimization purposes. The favorable option is to use metamodels for modeling the system. As mentioned previously, surrogate models provide an inexpensive means of representing the system response quantities. However, the cost associated with constructing a surrogate model increases dramatically as the number of design variables increases. Therefore one aspect of the methodology focuses on developing strategies through which the cost associated with the construction of a surrogate model can be reduced.

The basic paradigm in this research is to divide the global domain into local domains and construct accurate approximations over these local regions. The second paradigm is to efficiently perform design optimization over the surrogate models. The basic idea is to use the fast convergence properties of a gradient based optimization method locally. Optimization in different local regions ensures the exploration of the design space locally. In this way the whole design space is explored on a large scale via small scale searches in local regions. The process of model construction and optimization in the local regions can be performed on a parallel platform. This offers significant computational savings. Note that, the term parallel computing is loosely used in this thesis. The term parallel computing is used to refer to distributed computing where the computation on one computer is independent of the computations on rest of the computers on distributed network. The steps of the GO-LAO algorithm are described below:

1. **Division of design space:** The global design space is divided into a number of local regions. For small design variable problems (up to 4 variables), the domain is divided into two parts along each design variable. The number of

local regions generated by this method is:

$$NL = 2^{n_{dv}} \quad (5.1)$$

where n_{dv} is the number of design variables and NL is the number of local regions. However the number of local regions grows exponentially with the number of design variables. Thus for problems with a large numbers of design variables, it is difficult to manage a large number of local regions. The following heuristic has been developed to address this problem.

Initially, four randomly chosen design variables are divided into two domains. This results in 16 local regions and Kriging models are constructed in each of them. The strategy for further division of the design space depends on the correlation parameters of the Kriging model. Recall from Chapter 2 that the correlation parameter θ 's indicate how local the estimate is. For a particular Kriging model, if the correlation parameter (θ_k) along k th design variable is small, then points farther away influence the model. Thus the design space along k th variable is divided into two parts. This way, the domain is further divided along those variables which have a low correlation parameter. Future research is required to address the curse of dimensionality.

- 2. Construction of approximation:** In each local region, design sites are sampled to fit the surrogate model. Space filling samples are generated by using a design of experiments (DOE) to capture all the interactions of the system variables. The Kriging approximation is constructed in each of the local regions. Initially the approximation is constructed by using a minimum number

of sample points. The number of sample points are chosen as the minimum number of points required for construction of a quadratic approximation. The minimum number of points are:

$$n_s = \frac{(n_{dv} + 1)(n_{dv} + 2)}{2} \quad (5.2)$$

where n_{dv} is the number of design variables and n_s is the number of sample points. Because a Kriging model in one region is independent of the others, the model in each of the local regions can be constructed in parallel [39]. It will reduce the model construction time and will provide significant time savings.

3. **Model Validation:** Each Kriging model is assessed for quality in terms of the cross-validation statistic $R_{prediction}^2$. This measure of accuracy given in Eq. 2.15 is used in this investigation. In this research, a $R_{prediction}^2$ value less than 0.9 is considered poor. If the Kriging model error of prediction is poor, then information from additional sample points is used to re-construct the new Kriging model. This step of increasing the fidelity of the approximation by adding more sample points will be referred to as “*refinement by addition*” in the following text. The fidelity of the Kriging models is improved through successive refinement by addition. Refinement by addition is a heuristic based approach. The number of additional sample points is decided by how bad the approximation is. If the calculated error is not too bad then a small number of points are added. However if the approximation is very bad then a large number of sample points are added. The number of additional points depends on several factors like the computational resources at one’s disposal, expense of the function evaluation and the desired accuracy. If the exact function

evaluation is very costly then additional points are added in small number. If the cost is moderate then relatively large number of sample points can be added. Following scheme is used to improve the accuracy:

$$\text{If } 0.5 \leq R_{prediction}^2 < R_{tolerable}^2 \quad (5.3)$$

$$n_{as} = n_{dv} + 1$$

$$\text{If } R_{prediction}^2 < 0.5$$

$$n_{as} = 2(n_{dv} + 1)$$

where $R_{tolerable}^2$ is the user specified tolerable accuracy and n_{as} is the number of additional sample points.

It should be noted that the addition of points is a heuristic based approach. New Kriging models are then re-constructed using the information from additional points and the error is re-calculated. This is an iterative step which is carried out until the quality of the approximation is improved to a desired level. In this way accurate Kriging approximations are constructed through error driven model management.

4. **Local optimization:** The next step after constructing the Kriging models is to perform optimization. Gradient based optimization is performed in each of the local regions. The local optimizers guarantee efficiency and speed in producing local optimum solutions. The following optimization problem is solved for each of the local regions.

$$\begin{aligned} \text{Minimize:} & \quad \tilde{f}(\mathbf{x}) \\ \text{Subject to:} & \quad \tilde{g}_i(\mathbf{x}) \leq 0 \quad i = 1, \dots, m \end{aligned} \quad (5.4)$$

$$\begin{aligned}\tilde{h}_j(\mathbf{x}) &= 0 & j &= 1, \dots, n \\ \mathbf{x}^l &\leq \mathbf{x} \leq \mathbf{x}^u\end{aligned}$$

A multi-starting point strategy is applied to take care of multiple local optima in a particular local region. An n_{dv} number of random initial starting points are selected to perform local optimization in each of the local region, where n_{dv} is the number of design variables. Note that some starting points will lead to the same optimum point. This multi-starting point strategy tries to capture different local optima. Thus by solving the optimization problem in Eq. 5.4 for n_{dv} starting points, one expects to capture the different local optima in a particular local region. It should be noted that one is not guaranteed of finding all the local optima. Again the multi-starting point strategy can be applied on a parallel platform. This will further enhance the time saving benefits.

5. **Global optimum solution:** After each of the local regions has been optimized, the individual local optimum neighborhoods are further analyzed. A reduced trust region (*i.e.*, neighborhood) around each local optimum is refined and used to locate exact local optimum solutions. To obtain the exact local optimum, the fidelity of a model in the trust region around each local optimum is improved through refinement by addition. The function value at the local optimum obtained from the Kriging model is compared with the exact function value. The size of the trust region is calculated based on the error between actual function and Kriging prediction at the local optimum, as shown

below:

$$\text{trust factor} = t.f = \frac{|f_k^* - f_e^*|}{|f_e^*|} \times 100\% \quad (5.5)$$

where f_e^* is the exact function value and f_k^* is the predicted value from Kriging model at the local optimum. The size of the trust region in any local region is calculated as:

$$\text{If } t.f \geq 100\% \quad (5.6)$$

$$\mathbf{x}_{tr}^l = \mathbf{x}^l$$

$$\mathbf{x}_{tr}^u = \mathbf{x}^u$$

$$\text{If } t.f < 100\%$$

$$\mathbf{x}_{opt}^l = \mathbf{x}_{opt} - \left| \frac{t.f(\mathbf{x}^l + \mathbf{x}^u)}{2} \right|$$

$$\mathbf{x}_{opt}^u = \mathbf{x}_{opt} + \left| \frac{t.f(\mathbf{x}^l + \mathbf{x}^u)}{2} \right|$$

$$\mathbf{x}_{tr}^l = \mathbf{x}_{opt}^l$$

$$\mathbf{x}_{tr}^u = \mathbf{x}_{opt}^u$$

$$\forall k = 1, \dots, n_{dv}$$

$$\text{If } x_{opt}^l(k) < x^l(k)$$

$$x_{tr}^l(k) = x^l(k)$$

$$\text{If } x_{opt}^u(k) > x^u(k)$$

$$x_{tr}^u(k) = x^u(k)$$

where \mathbf{x}_{opt} is the local optimum, \mathbf{x}_{tr}^l is the lower bound of the trust region and \mathbf{x}_{tr}^u is the upper bound of the trust region. The reduced trust region is always contained in the original local region.

The fidelity of the Kriging models in the trust region is improved through refinement by addition. A multi-start optimization is performed in this trust region and a new trust region is calculated based on the difference between exact function value and Kriging prediction at new local optimum. This process is iterated until the Kriging prediction matches closely with the exact function value or the size of the trust region is reduced to 10% of the local region, which is a heuristic based choice. The exact local optimum solution is obtained by re-solving the optimization in the final trust region. This heuristic model management strategy is used to drive the design to the exact local optimum. These exact local optimum solutions are compared to provide possible global optimum solutions.

The steps of the GO-LAO algorithm have been explained above in detail. It will be useful to look at the main steps through a flowchart. A schematic of the flowchart of the algorithm is given in Fig. 5.1.

5.4 Illustration of the GO-LAO Algorithm

The important steps of the GO-LAO algorithm are illustrated in this section with the help of a two design variable function. The two-dimensional analysis imparts a vivid visualization of the design space. The two variable Camelback function described in Sec. A.1 is a highly non-linear and multimodal function. It can provide a clear understanding of the algorithm and thus was chosen to illustrate the steps of the GO-LAO algorithm.

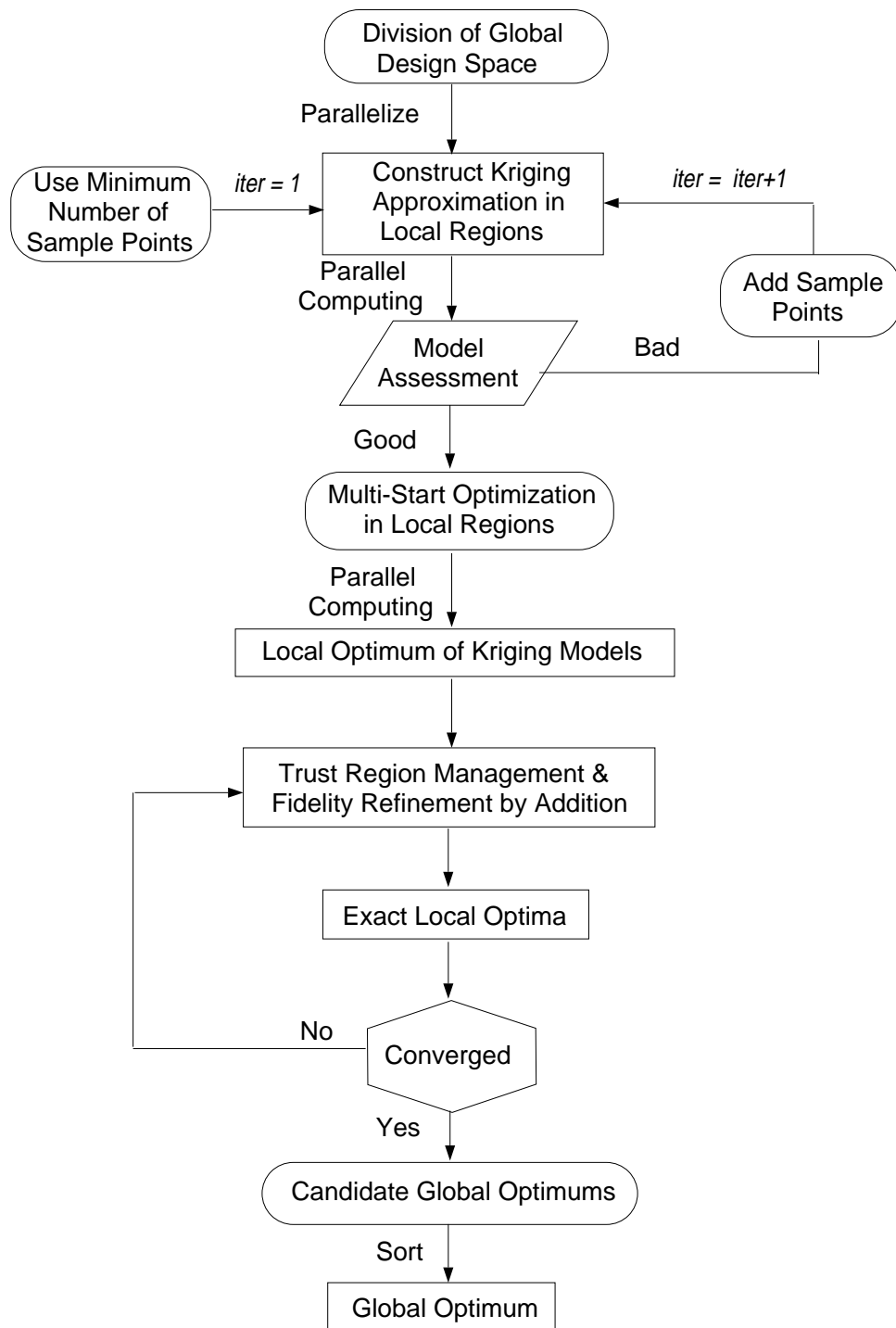


Figure 5.1. Flowchart of GO-LAO Algorithm

5.4.1 Division of Space

The global design space of the camelback function is divided into 4 regions by dividing the space along each of the variables into 2 parts. A schematic of the partitioning of the design space into 4 local regions is shown in Fig. 5.2. The figure also shows the sample points used to quantify function behavior in each of the local regions. The space filling samples were selected using a uniform design sampling scheme taken from the website <http://www.math.hkbu.edu.hk/UniformDesign>. These uniform samples employ efficient fractional factorial design and were developed with the objective of generating uniformly distributed points in the design space [10].

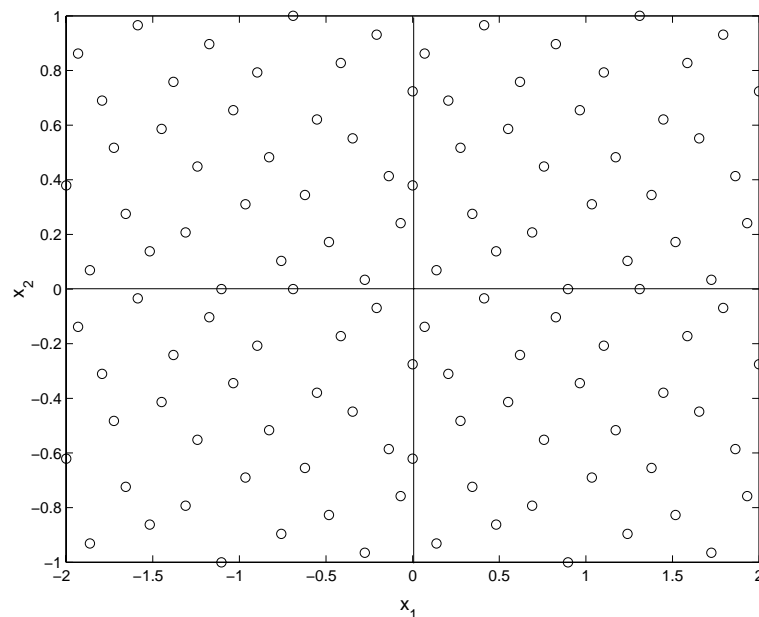


Figure 5.2. Division of the Global Design Space of the Camelback Function

5.4.2 Constructing Accurate Approximations

Construction of accurate approximations based on sample size is illustrated in Figs. 5.3- 5.7. The exact surface plot of the Camelback function is shown in Fig. 5.3. The Kriging approximation was constructed in each of the 4 local regions. Initially, a minimum number of sample points are used to construct the approximation. As discussed earlier in Eq. 5.2, the minimum number of points required to construct a quadratic approximation for a 2 design variable function are:

$$n_s = \frac{(2 + 1)(2 + 2)}{2} = 6. \quad (5.7)$$

Thus 6 sample points in each of the 4 local regions are used to construct the Kriging approximation in Fig. 5.4. Overall 24 sample points are used to construct the Kriging model for the Camelback function. As illustrated in Fig. 5.4, the quality of the approximation is extremely poor. After adding 14 additional sample points in each region, the accuracy of the approximation is still unacceptable. A Kriging approximation with a total of 80 sample points (20 in each region) is shown in Fig. 5.5. The quality of the approximation improves with the addition of more sample points, as shown in Fig. 5.6. Fig. 5.7 illustrates the Kriging approximation for 120 sample points, which is equivalent to 30 points in each of the 4 regions. Thus improved approximations can be obtained by adding information from additional sample points. Fig. 5.8 plots the $R_{prediction}^2$ measure of accuracy versus the number of sample points. This illustrates the improvement in the quality of the approximation through additional sample points.

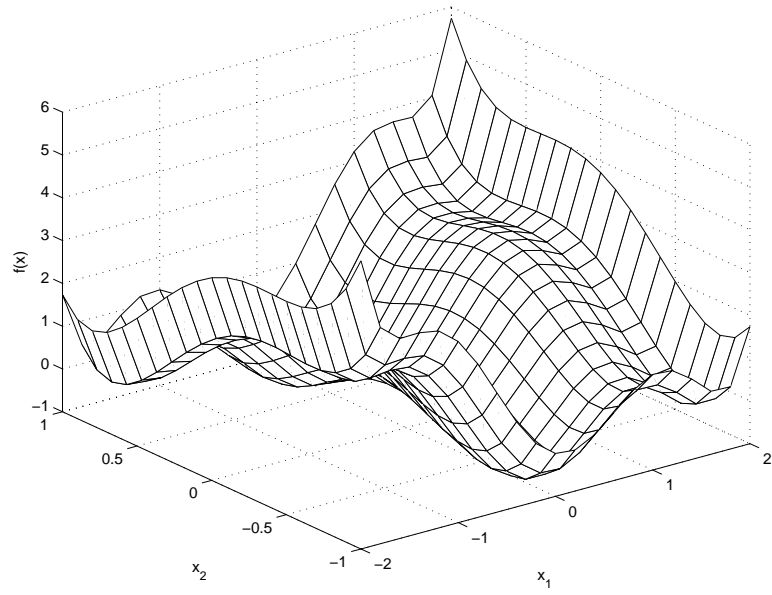


Figure 5.3. Surface Plot of Exact Camelback Function

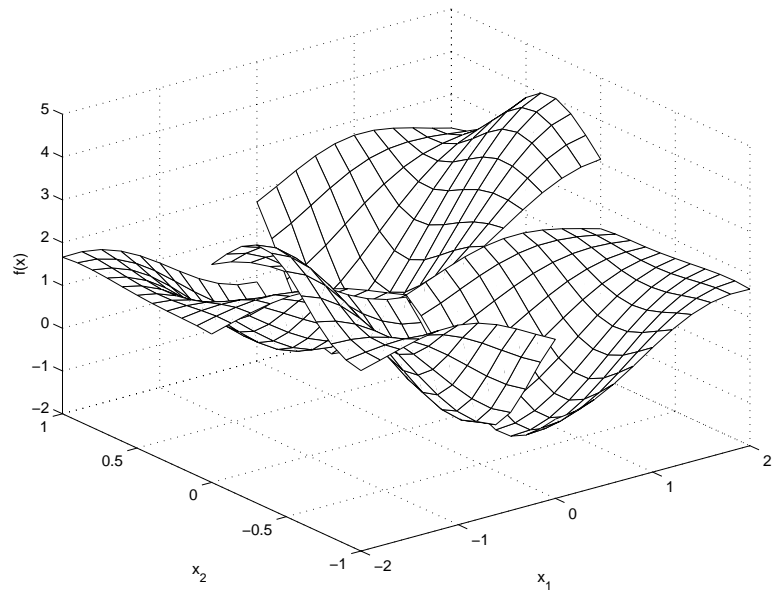


Figure 5.4. Kriging Approximation of Camelback Function with Minimum (24) Sample Points

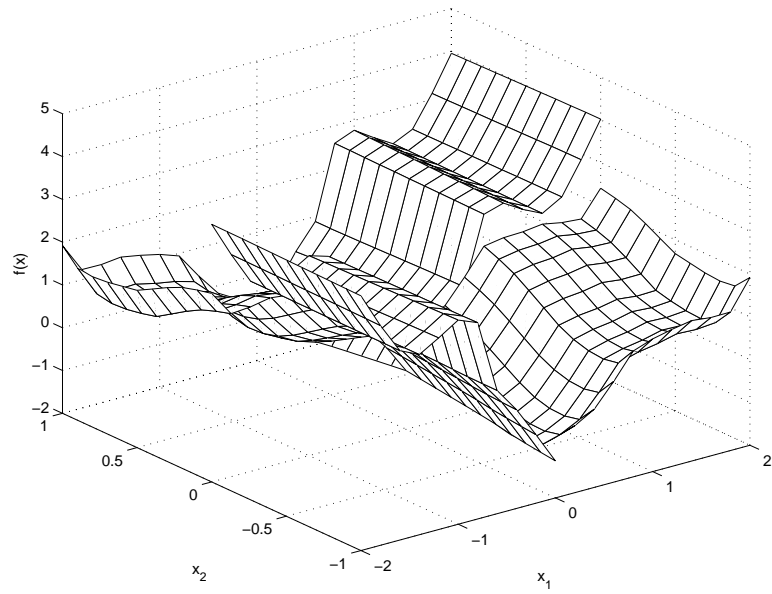


Figure 5.5. Kriging Approximation of Camelback Function with 80 Sample Points

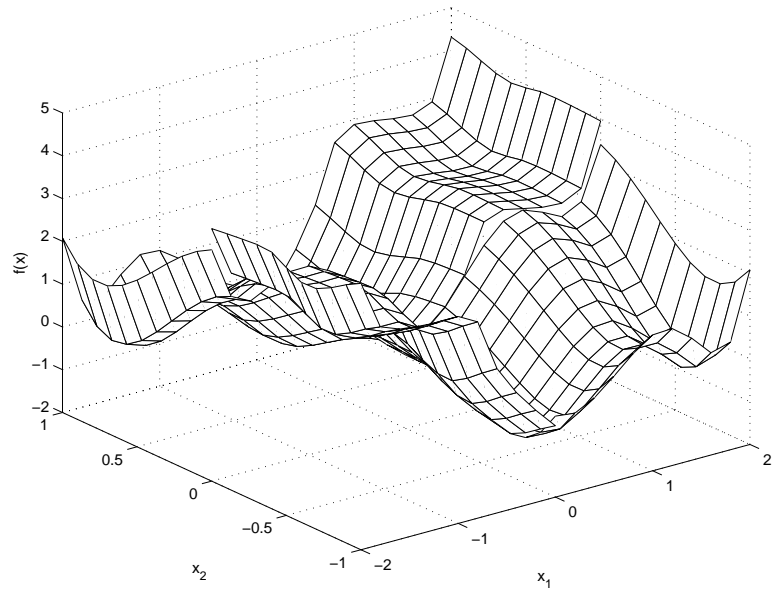


Figure 5.6. Kriging Approximation of Camelback Function with 104 Sample Points

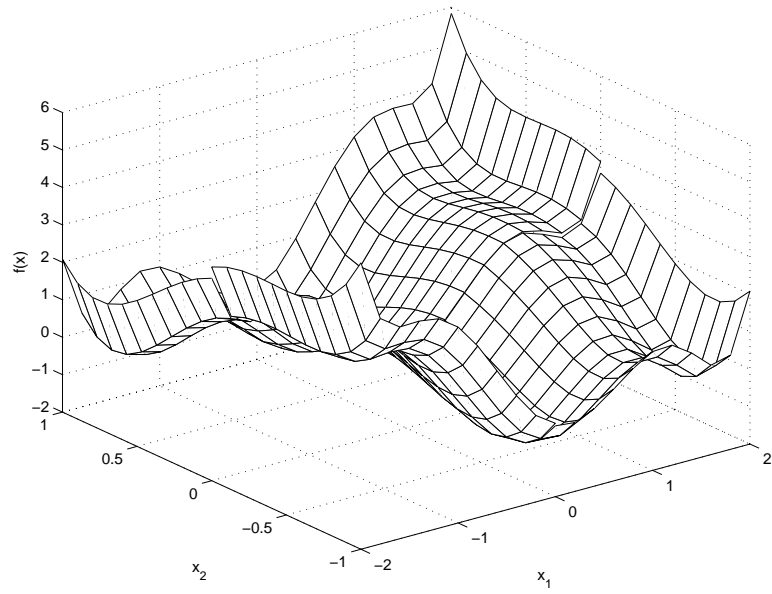


Figure 5.7. Kriging Approximation of Camelback Function with 120 Sample Points

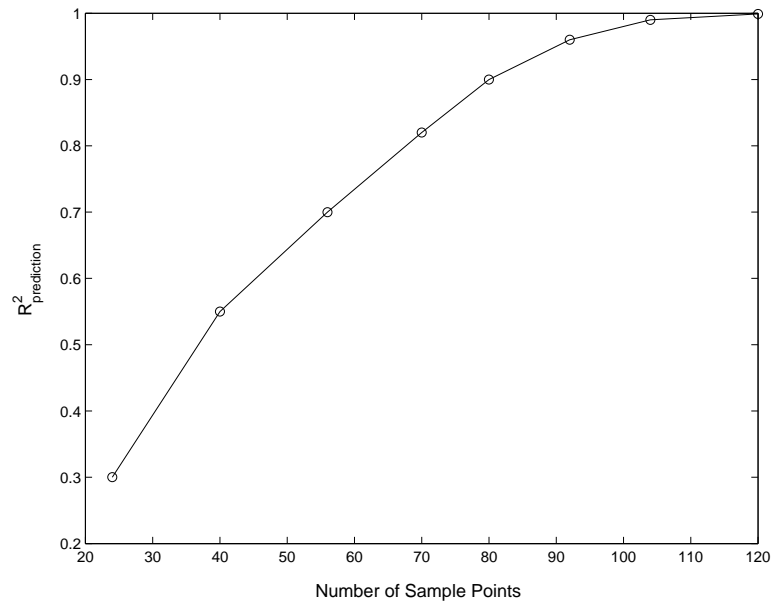


Figure 5.8. Improvement in Accuracy with Additional Sample Points

5.5 Advantages of GO-LOA

The proposed global optimization method has the following advantages:

1. For any problem, the time required for constructing the approximation over a global domain would be large as compared to the time required for constructing the local approximations. The large amount of time required for constructing a Kriging model over a global domain is due to the sub-optimization problem in Eq. 2.7 for evaluating Maximum Likelihood Estimates (MLE). It should be noted that the sub-optimization problem is an n_s (number of sample sites) dimensional problem. Hence it is evident that the greater the number of sample points, the larger the time required to construct the Kriging model. Thus computational time can be saved by reducing the computational effort required in the Kriging sub-optimization problem. This can be accomplished by constructing the surrogate models over small local regions. As one local model is independent of the other, the model construction process can be easily implemented in parallel [39]. Thus, by constructing the local approximations in parallel, the model construction cost can be greatly reduced.
2. After local approximations are constructed, they can be employed in efficient design optimization over each of the local regions. The savings in computational time through any avenue are desired in this fast paced world. The savings in time are augmented by performing optimization over local regions in parallel. In a particular local region, the optimization can be performed simultaneously from all the multi-starting points. Grid computing could be

taken advantage of for this type of work. The time saving benefits are not only achieved by simultaneously performing multi-start optimization in a particular region but also through parallelizing the optimization processes across the different local regions.

3. The GO-LAO uses gradient based local optimizers. Hence the available gradient information is also utilized in this method. The feature of fast convergence of gradient based optimization methods is exploited to quickly achieve local optimal solutions.

5.6 Test Problems

The *global optimization method using local approximate optimization* (GO-LOA) has been tested on three test problems. The problems are chosen such that the global optimization methods are susceptible to being stuck in local optima. A comparison with genetic algorithms (GA) is also made to assert the advantages of the GO-LAO method. The results are presented in the following sub-sections.

5.6.1 Camelback Six-Hump Function

The formulation of the test problem, Camelback Six-Hump function is given in Sec. A.1. This is a two design variable problem which is highly non-linear and multimodal. It has six local optima and two global optimum solutions. This function was used in Sec. 5.4 for illustrating different steps of the algorithm. A plot of the design space is shown in Fig. 5.3. A schematic of the contour plot of the function

is given in Fig. 5.9. The exact global optimum solutions of this problem are given in Table 5.1. Note that there are two global optimum solutions.

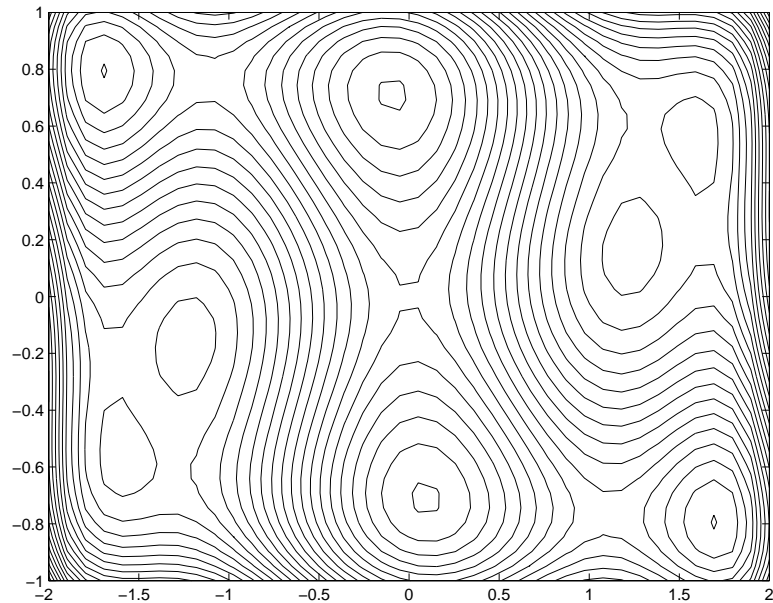


Figure 5.9. Contour Plot of the Camelback function

In this study, the exact function analysis was used to find the optimum solution by using a Genetic Algorithm. In the GA implementation, a roulette wheel selection method is employed by using 10 members as the population size. A single-point crossover is used for the crossover operation. A linearly decreasing mutation rate is used, which decreases from 0.14 to 0. The elitism approach is used to save the fittest member of the population and is copied to the next generation. The elitism approach ensures that the future generations have better or at least the same fit member. The GA is stopped when the variance of each of the design variable in a particular generation becomes smaller than a tolerance value. The tolerance in variance is the maximum allowable error in variance of each of the design variables

TABLE 5.1

EXACT GLOBAL OPTIMUM SOLUTIONS OF THE CAMELBACK
FUNCTION

Optima #	Global Optimum Designs (x^*)	Objective Function Value (f^*)
1	(0.089, -0.713)	-1.030
2	(-0.089, 0.713)	-1.030

for a particular generation. In other words, the GA is stopped when all the members of a generation become the same within an error which is less than the tolerance. Due to the random nature of GA, several runs were performed.

Table 5.2 shows the number of generations, function evaluations, optimum design and CPU time for different trials of the GA. A total of 15 trials were performed for different tolerances in the variance. In this table, a cross (X) implies that the optimum is not obtained and \approx means that the obtained optimum is close to the actual. In a conventional GA, the number of function calls are equal to number of generations multiplied by the population size. Thus even for the successful 3rd trial, with the smallest number of generations among successful trials, the number of function calls would be 120. However in this GA study, all the members of a particular generation were checked if they have already been evaluated. If they are already evaluated then the function value at that particular member is directly

read from a table compiled from generation to generation. This is the reason that in Table 5.2, number of function calls are less than the product of the number of generations and population size. One may be surprised to see such a small difference between the number of function calls and the generations. The reason for such a small difference is that the most fit member keeps on duplicating itself after a few initial generations and thus for future generations, the design does not change much. Hence the value of those repeated members is read from the compiled table. Note that this is not usually done in a GA. This approach was implemented to reduce the number of function evaluations.

The increasing trend in CPU time with decreasing tolerance is clearly evident from Table 5.2. For relatively large tolerance values as shown in trial 1 and 2, the GA could not identify the exact optimum. Out of 15 simulation runs, the GA could not locate the optimum in 4 trials. Small values of tolerance does not guarantee convergence to exact optimum solutions. The GA could not locate the optimum even for small tolerances of 10^{-5} and 10^{-7} , as shown in Trials 5 and 15. Thus in some cases, it is also possible that the GA produces incorrect results. The randomness in the GA is clearly evident from the large difference in the number of function evaluations required for the same level of tolerance. It should be noticed that for the tolerance level of 10^{-6} , the GA required 69 and 182 function evaluations in two different trials. Hence the GA may require extra function calls in different runs for the same set of parameters. Thus for some cases, it may require unnecessary function evaluations, thereby increasing the computational cost. The GA was able

TABLE 5.2

DIFFERENT GA TRIAL RUNS FOR CAMELBACK FUNCTION

Trial #	Tol. in Var.	# of Gen.	# of F-Calls	Optimal Design	f^*	CPU Time (sec)	Optima #
1	10^{-3}	8	29	(-0.0431, 0.5538)	-0.867	1.94	X
2	10^{-3}	16	52	(-0.0978, -0.667)	-0.8846	2.19	X
3	10^{-4}	12	55	(0.0978, -0.7104)	-1.0173	2.91	≈ 1
4	10^{-4}	34	76	(0.0978, -0.7613)	-1.0314	4.3	≈ 1
5	10^{-5}	23	65	(0.137, -0.8317)	-0.893	2.48	X
6	10^{-5}	39	78	(0.0978, -0.7434)	-1.0173	5.47	≈ 1
7	10^{-5}	52	104	(-0.09, 0.7691)	-1.0035	9.83	2
8	10^{-6}	35	69	(0.0978, -0.7104)	-1.0132	4.22	≈ 1
9	10^{-6}	57	137	(-0.09, 0.7143)	-1.0316	14.47	2
10	10^{-6}	71	101	(0.0978, -0.7434)	-1.0173	10.27	≈ 1
11	10^{-6}	92	182	(0.09, -0.7143)	-1.0316	18.67	1
12	10^{-7}	39	73	(0.09, -0.7143)	-1.0316	4.39	1
13	10^{-7}	44	105	(0.09, -0.7143)	-1.0316	9.08	1
14	10^{-7}	63	138	(0.0822, -0.6556)	-1.0072	11.23	≈ 1
15	10^{-7}	90	142	(0.0039, 0.6869)	-0.9941	14.61	X

to capture the second optimum in only 2 trial runs out of 11 successful trials. Hence there is a probability of missing some of the multiple global optima.

While employing the GO-LAO algorithm, the global domain is divided into 4 local regions having equal area. The sampling sites in the design space are shown in Fig. 5.2. For constructing the Kriging model, all the initial parameters in the correlation function were set to 0.5. The desired level of accuracy $R_{tolerable}^2$ was taken to be 0.9999. Each of the 4 local regions required 30 function evaluations. A total of 120 function evaluations were required to build the Kriging approximation with $R_{tolerable}^2 = 0.9999$. Refinement in the trust region is not required in this case. The global optimization results are tabulated in Table 5.3. The table provides a comparison between the GO-LAO approach and the genetic algorithm (GA). The GA results shown here are the trials 9 and 10 in Table 5.2. As can be seen that the global optimum solution is located within a very small error by the GO-LAO. The GA also locates one of the global optimum solution with a smaller error than GO-LAO. However the GA is not able to locate the other optimum in most of the simulation runs. The GO-LAO requires 120 exact function calls and the GA requires 109 calls, which is the average of the function calls in trials 6 to 14. Thus the computational cost of GO-LAO approach is comparable to GAs. At almost the same computational cost, the GO-LAO strategy is able to locate both of the global optimum solutions with good accuracy. However the GA fails to locate one of the optimum solutions most of the time. Thus the GO-LAO strategy presents an efficient framework for global optimization.

TABLE 5.3

GLOBAL OPTIMIZATION RESULTS FOR THE CAMELBACK FUNCTION

Approach	Global Optimum Design (\mathbf{x}^*)	Global Optimum f^*	Error % in \mathbf{x}^*	Error % in f^*	# of F-Calls
Exact	(0.089, -0.713)	-1.030	-	-	-
	(-0.089, 0.713)	-1.030	-	-	-
GA	(-0.09, 0.7143)	-1.0316	0.23	0.15	137
	(0.0978, -0.7434)	-1.0173	4.4	1.23	101
GO-LAO	(0.095, -0.714)	-1.0315	0.84	0.14	120
	(-0.095, 0.714)	-1.0315	0.84	0.14	120

A plot of the convergence histories for the different multi-start optimizations in 4 local regions for the GO-LAO strategy is shown in Fig. 5.10. Optimization from two multi-starting points leads to the global optimum in regions 1 and 4. The optimization in region 2 produces a local optimum and the optimum in region 3 reaches the variable bounds. A convergence plot for the genetic algorithm for 9th trial is given in Fig. 5.11. The number of function evaluations for GA are the function calls to the exact system analysis and thus contribute to the computational cost. On an average the GA implementation required 109 exact function analysis calls. It should be noted that the number of function evaluations in GO-LAO are the evaluations of the Kriging approximations in the local regions. The number of function calls shown in Fig. 5.10 does not contribute to the system analysis calls. The small number of function calls to the Kriging models for GO-LAO should not be confused with exact system analysis calls. The function calls for GO-LAO come from construction of accurate approximations which amounts to 120 evaluations (*i.e.*, 30 samples in each local region).

The CPU time for GO-LAO approach and GA is compared in Fig. 5.12. The local modeling time is the maximum of all the CPU times required to construct accurate models. As the Kriging models are constructed in parallel, the local modeling time is the bottleneck of individual modeling times. Similarly the local optimization time is the bottleneck of the optimization performed in local regions. The total CPU time for GO-LAO approach comes out to be 1.5 *sec* and 9.7 *sec* for GA. It should be noted that the CPU time for GA varies from trial to trial. Thus the time reported here is the average of time in trials 6 to 14. In this study, a true parallel

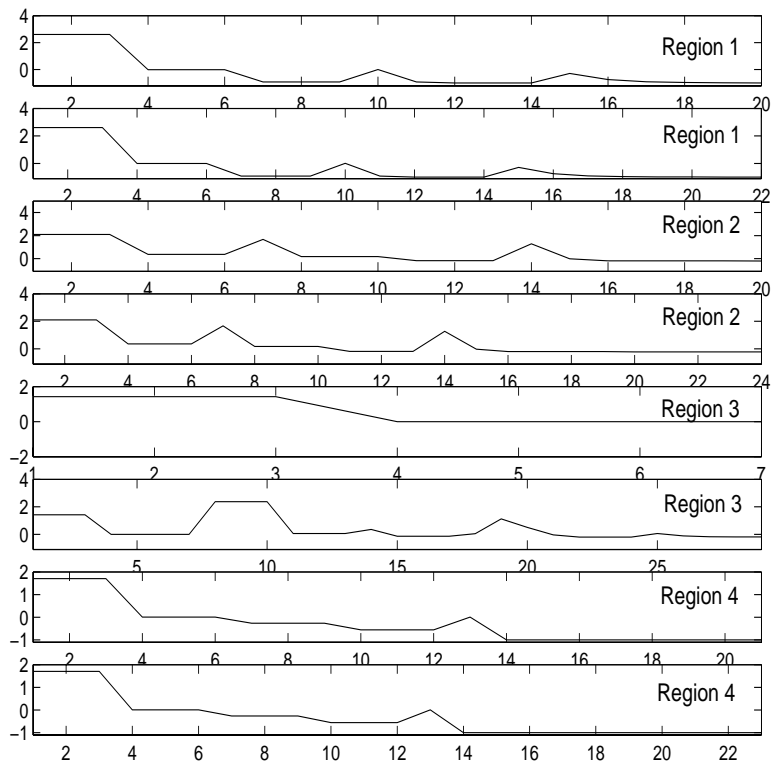


Figure 5.10. Convergence Histories in GO-LAO for Camelback function

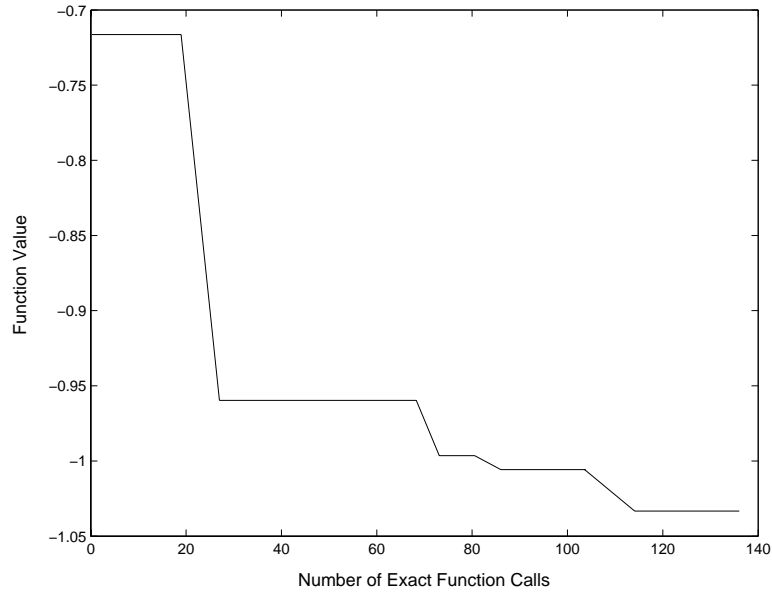


Figure 5.11. Convergence History in GA for Camelback function

evaluation was not performed for GO-LAO. In fact, local regions were sequentially processed and the bottleneck of individual times is reported as if they were computed in parallel with no communication overhead. There is a communication delay in actual parallel computing environments, which has not been taken into account in this investigation. Hence the time for GO-LAO will be a little more than 1.5 *sec*. The total potential savings in CPU time is close to 84%. It is possible to reduce the CPU time in GA by parallelizing the genetic operations on different members of a particular generation. Although parallel processing is possible in a GA, the evaluation of the new generation depends on the result from previous generation. Hence a complete parallel platform is not possible in GA.

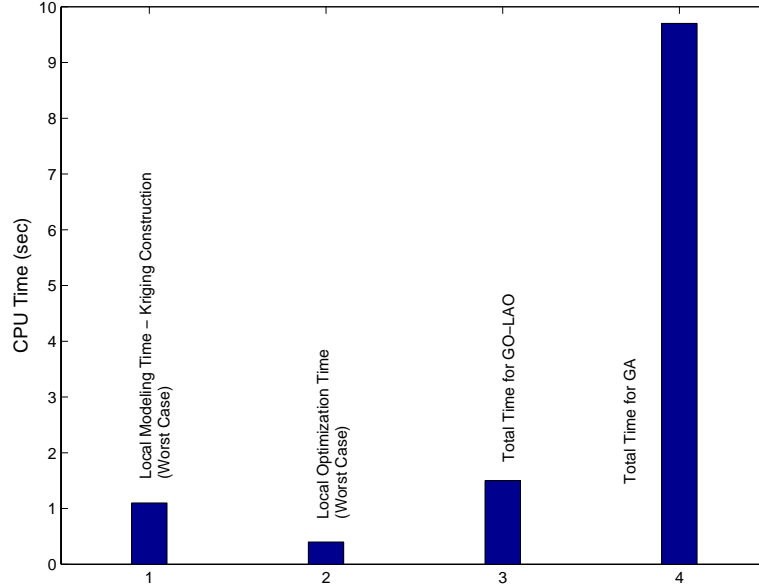


Figure 5.12. Computational Time for the Camelback function

5.6.2 Engine Design Problem

This problem deals with design of the geometry of an engine. The compound valve geometry is shown in Fig. A.1. The problem is described in Sec. A.2. The objective in this problem is to maximize the brake power. It is a constrained problem with 8 design variables, 2 equality constraints and 10 inequality constraints.

In GA implementations, a population size of 40 members was used with 10^{-4} as the tolerable error in variance. Other parameters in GA *i.e.* crossover method, mutation rate and selection procedure are the same as used in the Camelback problem. The initial population in GA is randomly created. A penalty approach is used to handle the constraints in this problem. As the GA does not handle constraints explicitly, the constraints are included in the fitness function by using a penalty

approach. The fitness function in the GA is calculated as:

$$fitness = f + \rho \left(\sum_i |g_i| \right) \quad (5.8)$$

where f is the objective function and the summation is over all the violated constraints. The penalty parameter (ρ) should be chosen carefully such that sufficient weight is given to the violated constraints. Three different values of penalty parameter (ρ) were used in this problem. The results from different trials of GA are tabulated in Table 5.4. A trial with $\rho = 100$ produced the best results out of the three trials. In this problem, unlike the Camelback problem, the function value were re-evaluated for any duplicate points in a GA generation. This is done to show that conventional GA is computationally intensive and thus requires excessive function calls.

For the GO-LAO approach, the global domain is divided into 16 regions by dividing the domain along 4 randomly chosen design variables. Kriging approximations are constructed in each of these 16 local regions and validated for a $R_{tolerable}^2 = 0.999$. These regions are further sub-divided into additional local regions by comparing the model parameters, as discussed in Sec. 5.3. At the end of the sub-division process, a total of 63 local regions were obtained. The remaining steps of GO-LAO algorithm were employed in these local regions.

The optimization results obtained from both the algorithms are compared in Table 5.5. The objective function value for the optimum obtained by the GA is less than that found by GO-LAO. However the optimum from GA does not satisfy all the constraints. The result from the GA violates both the equality constraint and two of the inequality constraints. Although the objective function value for the

TABLE 5.4

GA TRIALS FOR THE ENGINE DESIGN PROBLEM

Optimum Design	Exact	$\rho = 10$	$\rho = 100$	$\rho = 1000$
x_1	83.33	78.14	81.32	77.49
x_2	9.45	9.19	11.78	11.67
x_3	39.12	49.49	38.07	30.25
x_4	47.136	49.88	37.3	35.45
x_5	6.481	6.49	6.33	5.42
x_6	2.3×10^6	2.29×10^6	2.29×10^6	2.29×10^6
x_7	1.302	3.75	2.58	13.18
x_8	49.352	58.14	47.9	38.84
f^*	1.648×10^8	1.586×10^8	1.593×10^8	1.42×10^8
Generations	-	395	478	498
Function Calls	-	15800	19120	19920
% Error in \mathbf{x}^*	-	13.03	9.14	19.54
% Error in f^*	-	3.88	3.34	13.83

optimum obtained from GO-LAO is 14% less than the actual, it does satisfy all the constraints. The GA not only locates an infeasible solution but also requires a large number of function calls as compared to GO-LAO. The number of function calls in the GA trial are almost 3.4 times the evaluations in GO-LAO. The GO-LAO locates a feasible optimum candidate at a relatively low computational cost. The computational savings in system analysis calls offered by the GO-LAO are 70%.

5.6.3 Autonomous Underwater Vehicle (AUV) Design Problem

This problem focuses on the design of an autonomous underwater vehicle. A complete description of this problem is given in Sec. A.3. It is a 9 design variable problem with no constraints. This is an application that involves a more computationally intensive problem. Each function evaluation takes approximately 12 minutes on a Pentium IV 2.2 *GHz* Windows XP platform. The computational expense in this problem discourages the use of exact function analysis for optimization purposes. Hence one must resort to response surface approximations to reduce the computational cost.

In the GO-LAO approach, a value of $R_{tolerable}^2 = 0.9$ is used for this problem. The global domain is divided into 16 regions. Kriging model parameters are used to further divide these regions. Finally 27 regions are created which required a total of 1740 function evaluations. As this is a computationally intensive problem, the sub-division process is terminated after 2 divisions in a particular local region. A quasi-parallel computation was performed for this problem. Parallel computations

TABLE 5.5

GLOBAL OPTIMIZATION RESULTS FOR THE ENGINE DESIGN PROBLEM

Optimum Design	Exact	GO-LAO	GA ($\rho = 100$)
x_1	83.33	80.65	81.32
x_2	9.45	9.123	11.78
x_3	39.12	38.874	38.07
x_4	47.136	43.679	37.3
x_5	6.481	6.5	6.33
x_6	2.3×10^6	1.95×10^6	2.29×10^6
x_7	1.302	1.26	2.58
x_8	49.352	48.642	47.9
f^*	1.648×10^8	1.41×10^8	1.593×10^8
h_1	-0.0011	-0.0061	2.984
h_2	0	-0.0002	1.31
g_1	-0.0033	-2.683	-2.013
g_2	-6.405	-3.725	-4.395
g_3	0.0035	-0.091	-9.58
g_4	0.0062	-2.616	-8.11
g_5	-2.835	-0.0052	5.873
g_6	-0.101	-0.0739	-0.0102
g_7	-0.0002	-0.448	2.24
g_8	-0.02	0	-0.17
g_9	-0.01170	-0.0087	-0.039
g_{10}	-699.45	-739.94	-641.17
Function Calls	-	5670	19120

were performed on 5 different computers simultaneously. For the GA implementation, a population size of 90 members is taken. Other parameters for the GA *i.e.* crossover scheme, mutation rate and selection criteria are the same as used in other problems. It was run for 20 generations and it took 1800 function evaluations. This process took approximately 15 days to accomplish. The population size and the maximum number of generations for the GA are chosen such that the number of function evaluations are close to the number of evaluations required by GO-LAO. Here also, the function value at duplicate members was re-evaluated, as is done in a conventional GA.

The results obtained from the GO-LAO strategy are compared with the GA results in Table 5.6. The exact global optimum solution is not known for this problem. It is clearly evident that GO-LAO produces a better optimal design than the GA. The optimum solution found by GO-LAO is the best found so far. The GA locates an optimum solution which has objective function value greater than that found in GO-LAO. The optimum found by GO-LAO has a function value 31% less than that found by GA. Thus GO-LAO is able to lock the algorithm on to a potential global optimum solution. Moreover the cost of constructing accurate approximations is less than that required for exact function analysis in GA. Hence GO-LAO not only produces a better solution but is also able to locate it at a relatively lower computational cost. A plot illustrating the decrease in function value with the convergence in GO-LAO is shown in Fig. 5.13. A similar plot depicting the decrease for GA is shown in Fig. 5.14.

TABLE 5.6

GLOBAL OPTIMIZATION RESULTS FOR THE AUV PROBLEM

Optimum Design	Exact	GO-LAO	GA
x_1	Not Known	1.612	1.7825
x_2		13067	17406
x_3		1.061	1.0084
x_4		0.184	0.0485
x_5		0.7	0.888
x_6		0.213	0.0195
x_7		26.999	24.73
x_8		0.028	0.0132
x_9		0.012	0.0167
f^*	Not Known	-20.973	-14.451
Function Calls	-	1740	1800
CPU Time	-	7.8×10^4 sec \approx 1 day	128.3×10^4 sec \approx 15 days

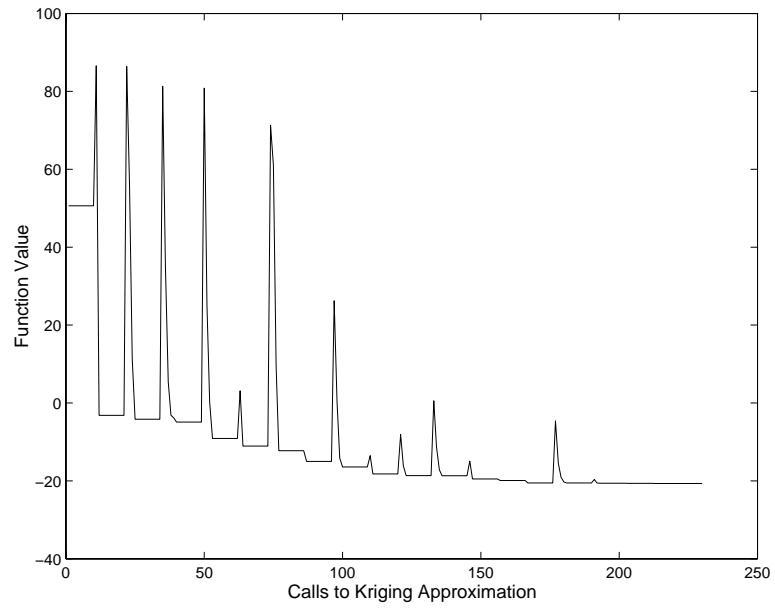


Figure 5.13. Convergence History in GO-LAO for AUV Problem

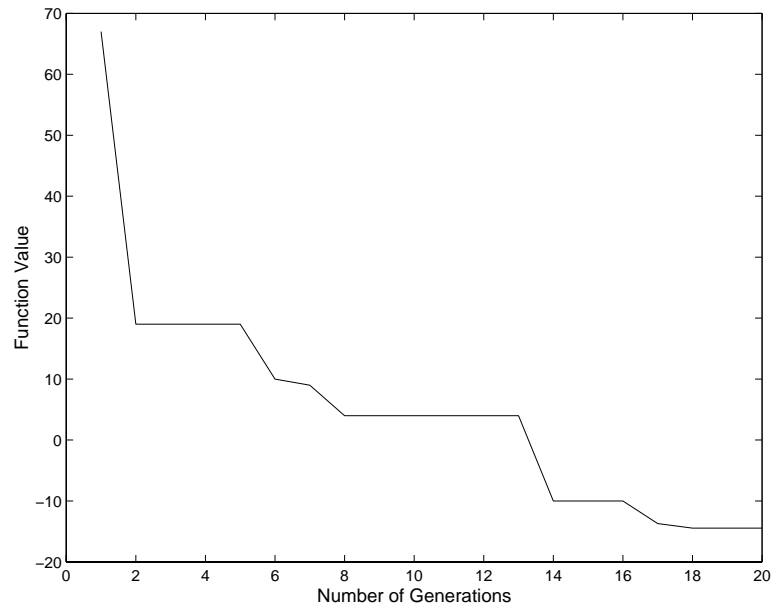


Figure 5.14. Convergence History in GA for AUV Problem

5.7 Summary

A strategy for global optimization based upon local approximate optimization has been developed. The strategy for the division of the global design space into small local regions is laid out. Kriging approximations are constructed in the local regions to provide an inexpensive formulation of the system behavior. The Kriging approximation is validated in terms of a cross-validation measure of error, to provide accurate mapping. Gradient based local optimizers are used to identify local optima in each of the regions. A reduced trust region around each local optima is further analyzed to provide the global optimum solution candidates.

Compared to genetic algorithms, the GO-LAO approach is not a random search. To search in the unexplored areas, some steps of the genetic algorithm operate by random nature. Thus unnecessary function evaluations are carried out, which adds to the cost of the process. However, savings in every possible avenue is desired in this cost-conscious world. In the proposed strategy, the Kriging model construction and optimization over the local regions can be performed in a parallel computing environment. This offers significant savings in computational time. Numerical simulation on selected test problems shows that this approach possesses the capability to identify candidate global optimum solutions. The advantages of the GO-LAO strategy can be summarized as:

- Savings in computational time by domain decomposition and parallel computing (for model construction and optimization).
- Identification of potential global optimum designs.

Thus the time saving benefits of the proposed approach can be exploited by the designer to arrive at global optimum designs within reduced time frames.

CHAPTER 6

NON-LINEAR POST-OPTIMALITY ANALYSIS

6.1 Post-Optimality Analysis

Post-optimality analysis is used to estimate variations in the optimum design when system parameter values are changed. Having obtained an optimum design, it is not uncommon for management to modify performance targets. These performance targets are fixed parameters in the optimization problem. Post-optimality analysis is used to understand the trade-offs resulting from these target modifications. Linear post-optimality analysis is restricted to linear sensitivity computations and can not provide us with an accurate estimate of the optimal point for large parameter changes [14, 45, 55]. These methods only provide us with the first order information about the change in objective function that will be incurred after introducing a change in the system parameter. In today's competitive world, any post-optimality analysis which captures the non-linearities in the system would be preferred. Thus a non-linear post-optimality strategy has been developed in the present work.

6.2 Linear Post-Optimality Strategy

The linear post-optimality analysis can be summarized as follows [14, 45, 55].

Let \mathbf{p} be the parameters in an optimization problem stated as:

$$\begin{aligned}
 \text{Minimize:} \quad & f(\mathbf{x}, \mathbf{p}) \\
 \text{Subject to:} \quad & g_i(\mathbf{x}, \mathbf{p}) \leq 0, i = 1, \dots, m \\
 & h_j(\mathbf{x}, \mathbf{p}) = 0, j = 1, \dots, n \\
 & \mathbf{x}^l \leq \mathbf{x} \leq \mathbf{x}^u
 \end{aligned} \tag{6.1}$$

The Lagrangian L is given by:

$$L = f(\mathbf{x}, \mathbf{p}) + \sum_{i=1}^m \mu_i g_i(\mathbf{x}, \mathbf{p}) + \sum_{j=1}^n \lambda_j h_j(\mathbf{x}, \mathbf{p}) \tag{6.2}$$

where there exists Lagrange multipliers λ_j for the j^{th} equality constraint h_j and μ_i for i^{th} inequality constraint g_i . Differentiating the Karush-Kuhn Tucker (KKT) optimality conditions with respect to parameter p_k results in the following linear system of equations [45, 55]:

$$\begin{bmatrix} \frac{\partial^2 L}{\partial \mathbf{x}^2} & \frac{\partial(\mathbf{h}, \mathbf{g})}{\partial \mathbf{x}} \\ \left[\frac{\partial(\mathbf{h}, \mathbf{g})}{\partial \mathbf{x}} \right]^T & \mathbf{0} \end{bmatrix} \begin{pmatrix} \frac{\partial \mathbf{x}}{\partial p_k} \\ \frac{\partial(\lambda, \mu)}{\partial p_k} \end{pmatrix} + \begin{pmatrix} \frac{\partial^2 L}{\partial \mathbf{x} \partial p_k} \\ \frac{\partial(\mathbf{h}, \mathbf{g})}{\partial p_k} \end{pmatrix} = \mathbf{0} \tag{6.3}$$

Only the active constraints are considered in the above formulation. Eq. 6.3 can be solved for the sensitivity of the design variables $\left(\frac{\partial \mathbf{x}}{\partial p_k} \right)$ with respect to the problem parameter p_k . The change in the optimum design point is given by:

$$\Delta \mathbf{x}^* = \Delta p_k \left. \frac{\partial \mathbf{x}}{\partial p_k} \right|_{\mathbf{x}=\mathbf{x}^*} \tag{6.4}$$

The sensitivity of the objective function with respect to the parameter p_k is given by[14]:

$$\frac{df^*}{dp_k} = \frac{\partial f^*}{\partial p_k} + \sum \mu_i \frac{\partial g_i^*}{\partial p_k} + \sum \lambda_j \frac{\partial h_j^*}{\partial p_k} \quad (6.5)$$

where star (*) represents the values at initial optimum and the summation is over the active constraints only.

This post-optimality analysis gives a linear estimate of the change in the optimum design and the optimum objective function. The linear post-optimality analysis has restricted use. The limitations of the linear analysis are:

1. The linear nature of the analysis fails to produce accurate results for non-linear problems, which is usually the case for MDO problems. Thus the linear analysis does not give an accurate estimate for non-linear problems.
2. It demands that the active constraints in the initial optimization should remain active after the post-optimality analysis. Thus the allowable amount of change in the parameter is such that the active constraints does not change.

6.3 Overview of Non-Linear Post-Optimality Analysis (NLPOA)

Optimization costs for solving MDO problems are often reduced by approximating the model of the system and then performing Sequential Approximate Optimization (SAO) [40, 46, 47]. Sequential approximate optimization uses local response surface approximations to arrive at the optimal solution without incurring large computational cost [35]. An effective framework is the interior-point trust-region sequential approximate optimization (IPTRSAO) approach developed by Pérez *et al.*

[40]. This algorithm uses local quadratic approximations to generate approximate feasible iterates. The case of an infeasible starting point is managed by constructing homotopies of the violated constraints [41].

This research effort makes use of the archival histories of sequential approximate optimization to develop a non-linear post-optimality analysis strategy. In this method, post-optimality analysis is performed by using the information gathered through the sequential approximate optimization process. The initial optimization is performed by using the IPTRSAO strategy [40]. This framework generates local RSAs at each iterate. These local RSAs capture the non-linearities in the individual local trust region. Thus the exact response of the system is conserved and provides non-linear attributes for the post-optimality analysis. Once the optimization is terminated, a cumulative response surface approximation is generated by blending the local RSAs, as discussed in Chapter 3. The Non-Linear Post-Optimality Analysis (NLPOA) is based on the cumulative response surface approximation of the objective function and the constraints. If a system parameter is altered, the new optimal design can be obtained by performing optimization over the CRSA without performing actual costly function evaluations. The proposed analysis gives a more accurate estimate of the new optimal design than the linear post-optimality analysis [14, 45, 55].

6.4 Mathematical Formulation

In this section, the mathematical formulation of the NLPOA is presented while revisiting some of the important concepts of CRSA. The formulation of cumulative

response surface approximation (CRSA) used here is different to a slight extent than that discussed in Sec. 3.4.

6.4.1 Cumulative Response Surface Approximation (CRSA)

Let NI be the number of design iterates visited during the sequential approximate optimization of the function $f(\mathbf{x})$. At each design iterate a local quadratic approximation is generated during the SAO process. These local quadratic response surface approximations $\tilde{f}_i(\mathbf{x})$ are archived for each design iterate \mathbf{x}_i and are represented as:

$$\tilde{f}_i(\mathbf{x}) = f(\mathbf{x}_i) + \nabla f(\mathbf{x}_i)^T(\mathbf{x} - \mathbf{x}_i) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_i)^T H_i(\mathbf{x} - \mathbf{x}_i) \quad (6.6)$$

where $\nabla f(\mathbf{x}_i)$ is the gradient of objective function evaluated at \mathbf{x}_i and H_i is the Hessian of the RSA. The cumulative response surface approximation (CRSA) of the function $f(\mathbf{x})$ is given by:

$$\hat{F}(\mathbf{x}) = \frac{\sum_{i=1}^{NI} \phi_i(\mathbf{x}) \tilde{f}_i(\mathbf{x})}{\sum_{i=1}^{NI} \phi_i(\mathbf{x})} \quad (6.7)$$

where the blending function $\phi_i(\mathbf{x})$ is the same as discussed in Sec. 3.4.

6.4.2 Nonlinear Post-Optimality Analysis (NLPOA)

The local RSAs capture the non-linearities in a local region. The local RSAs are used to construct the CRSA of the objective function and is represented as $\hat{F}(\mathbf{x})$. Thus the CRSA is able to represent the non-linearities of the system behavior by using the local information. In certain cases, the dependency of constraints on the design variables and the parameters can be isolated. The constraints could be

reformulated as the sum of a function of design variables \mathbf{x} and a function of system parameters \mathbf{p} :

$$g(\mathbf{x}, \mathbf{p}) = y(\mathbf{x}) + z(\mathbf{p}) \quad (6.8)$$

This is usually the case in problems where the constraint is set to be less than some specified upper limit. Take the example of structural problems where the total stress in the structural member should be less than some factor of yield stress. In this case, the constraint can be expressed as:

$$g(\mathbf{x}, \mathbf{p}) = \sigma_{total}(\mathbf{x}) - \frac{\sigma_{yield}}{f.s} \leq 0 \quad (6.9)$$

where $f.s$ is the factor of safety, total stress (σ_{total}) depends on the design variables \mathbf{x} and yield stress (σ_{yield}) is the system parameter \mathbf{p} . Similarly, in aerodynamic problems, the total lift on the airplane should be greater than the weight of the plane and the total thrust generated by the plane should be more than the drag on the plane. These constraints also fall in the above-discussed category. Hence this kind of constraints are commonly found and this analysis is valid for these type of constraints. This fact is utilized to construct the CRSA of $y(\mathbf{x})$ according to Eq. 6.7 and is represented as $\hat{Y}(\mathbf{x})$. A change in the system parameter results in new constraints. These new constraints can be approximated to be the sum of $\hat{Y}(\mathbf{x})$ and $z(\mathbf{p}')$. The new constraint at the modified parameter \mathbf{p}' is expressed as:

$$\hat{G}(\mathbf{x}, \mathbf{p}') \approx \hat{Y}(\mathbf{x}) + z(\mathbf{p}') \quad (6.10)$$

These CRSAs of new constraints are used along with the CRSA of objective function to perform the post-optimality analysis. The post-optimization problem can be

posed as:

$$\begin{aligned}
\text{Minimize:} \quad & \hat{F}(\mathbf{x}) & (6.11) \\
\text{Subject to:} \quad & \hat{Y}_i(\mathbf{x}) + z_i(\mathbf{p}') \leq 0, i = 1, \dots, m \\
& \hat{Y}_j(\mathbf{x}) + z_j(\mathbf{p}') = 0, j = 1, \dots, n \\
& \mathbf{x}^l \leq \mathbf{x} \leq \mathbf{x}^u
\end{aligned}$$

where \mathbf{p}' is the modified parameter. The optimization is performed on the CRSA of objective function subjected to new constraints to arrive at the new optimum design.

6.5 Advantages of NLPOA

The proposed method offers the following benefits:

1. The CRSA captures the non-linear behavior of the system response via the local quadratic approximations. The incorporation of nonlinear nature provides a better estimate of the post-optimal solution. Thus the proposed method presents a nonlinear post-optimality analysis.
2. The post-optimality analysis is performed over the inexpensive CRSAs instead of the actual costly high fidelity simulation codes. The use of approximations reduces the computational cost and provides an inexpensive formulation. Hence this analysis can be used in computationally intensive problems.
3. This strategy does not suffer from the limitation on active constraints. The initial active constraints need not remain active for this non-linear post-optimality analysis.

6.6 Test Problems

The non-linear post-optimality analysis (NLPOA) has been implemented on three test problems. The problems are chosen in a way to encompass a variety of complex MDO problems having small to large number of design variables. The results are presented in the following sub-sections.

6.6.1 Barnes Problem

The formulation of the test problem, Barnes function is given in Sec. A.4. This problem was chosen because it is a two design variable problem which is highly nonlinear. A plot of the design space is shown in Fig. 6.1. An infeasible starting

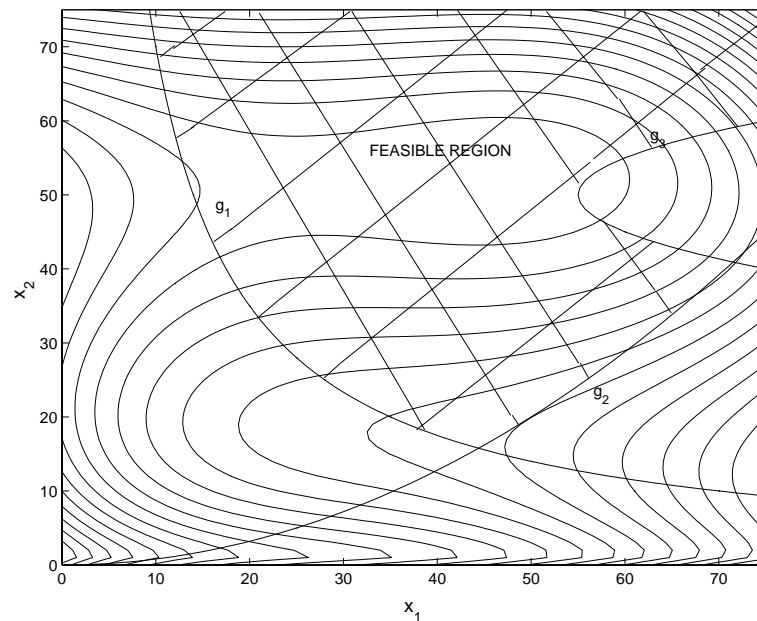


Figure 6.1. Design space of the Barnes problem

point $(65, 1)$ was used for the initial optimization. The interior point strategy (IP-TRSAO) was implemented to arrive at the optimal design point $\mathbf{x}^* = (49.47, 19.58)$. The algorithm converged in 7 iterations and the local RSAs were generated around the corresponding design iterates.

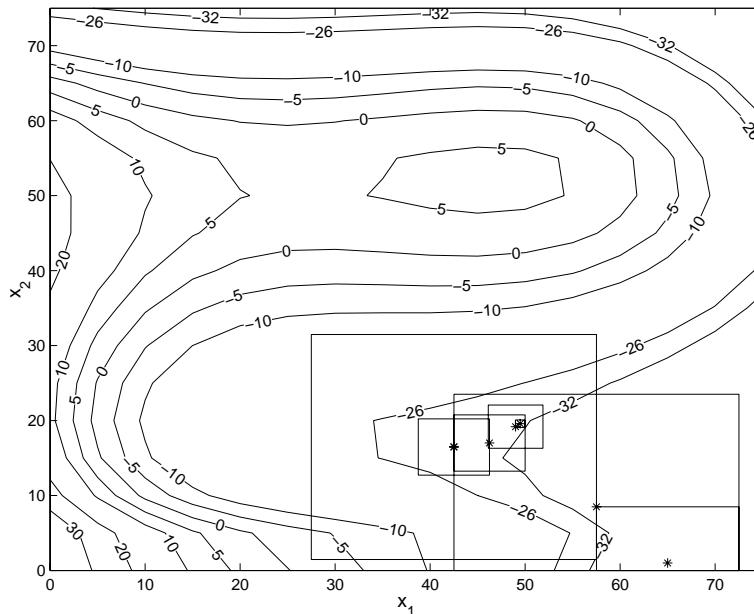


Figure 6.2. Exact contours of the Barnes problem

Fig. 6.2 shows the contours of the exact objective function. Contours of the CRSA for the Barnes function are shown in Fig. 6.3. The design iterates around which the local RSAs are built are shown by star (*) and their corresponding trust regions are indicated by rectangles around those points. The quadratic response surface approximation at each design iterate are used to generate the CRSA. It can be clearly seen from Fig. 6.3 that the CRSA is more accurate in the neighborhood of these iterates. All the design iterates and their corresponding trust regions lie in

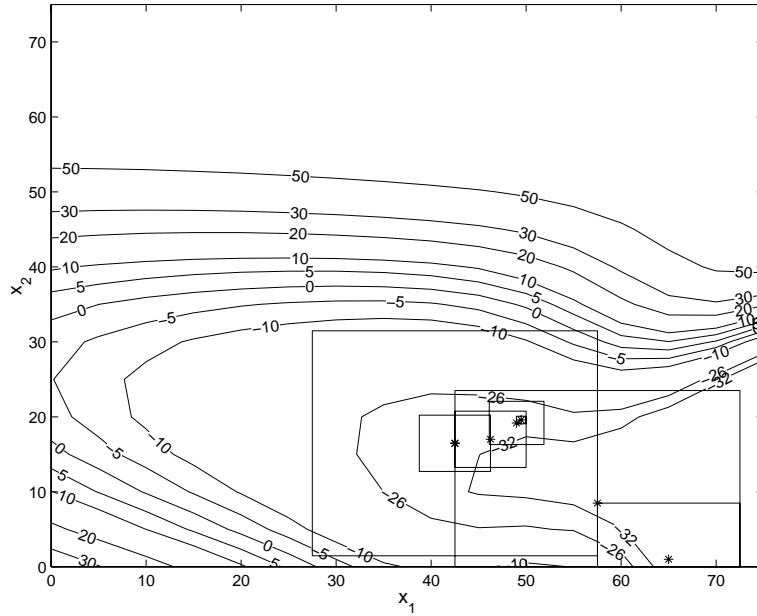


Figure 6.3. Contours of CRSA for the Barnes problem

the region defined by $25 \leq x_1 \leq 75$ and $0 \leq x_2 \leq 30$. Thus the CRSA gives a more accurate approximation in this region.

Contours of error percentage for the Barnes function are shown in Fig. 6.4. An error of only 0-5 % is incurred at the design iterates and overall an error of 0-10 % is encountered within the trust regions. The CRSA provides a good estimate of the actual function within the trust regions and can be used to perform the post-optimality analysis.

The post-optimality analysis was performed over the CRSA as in Eq. (6.11). The active constraint g_2 was changed from $g_2 \leq 0$ to $g_2 \leq 0.5$. The modified constraint is shown in Fig. 6.5 by the dotted line. The initial optimum point is shown by a cross (\times) and the new optimum corresponding to this change in the system parameter is shown by a triangle (Δ). The post-optimality results are shown in Table 6.1. The

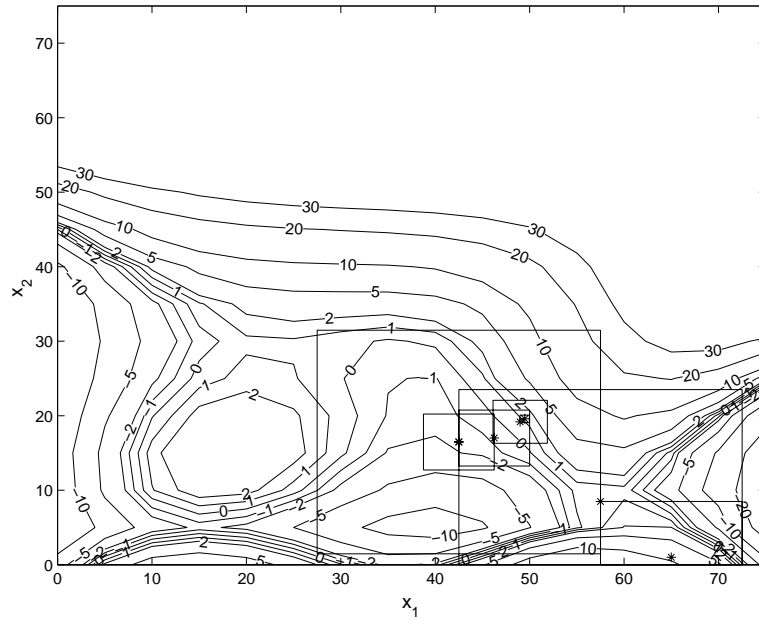


Figure 6.4. Contours of error percentage for the Barnes problem

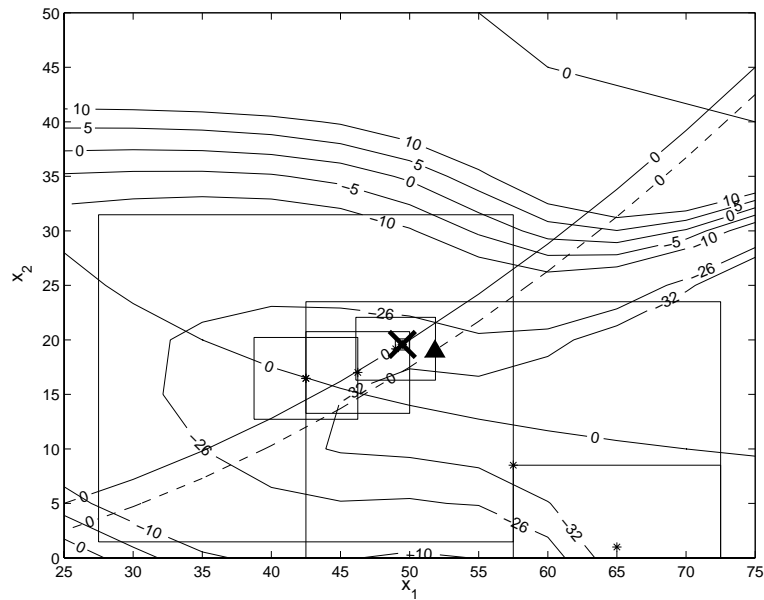


Figure 6.5. Post-optimality performed on the CRSA of the Barnes problem

table shows that a lesser error is experienced in the proposed CRSA-based approach as compared to the linear post-optimality analysis.

TABLE 6.1

POST-OPTIMALITY RESULTS FOR THE BARNES PROBLEM

Approach	Post-Optimal Design	Post-Objective Function	Max. error % in x	Error % in Objective Function
Exact	(52.67, 19.69)	-33.59	-	-
Linear Strategy	(48.25, 19.86)	-33.46	7.86	0.39
NLPOA	(51.83, 18.94)	-33.58	2.00	0.03

6.6.2 High Performance Low Cost Structure (HPLC) Ten-bar Problem

The physical problem of interest is a variable topology and payload version of a ten-bar truss. This was first introduced as an MDO test problem in Wujek [59]. The details of this 17 design variable and 13 constraints structural problem are given in Sec. A.5. In a structural problem like this, the important parameters are the material properties. A designer may be interested in observing the system behavior when the material of the structure is changed. This implies that the designer's interests lie in analyzing the post-optimal solutions obtained after changing the yield stress (σ_{yield}). In this test problem, the optimal system behavior has been studied owing to a change in the yield stress.

The optimization was run with the initial design of $L_i = 360$ in., $M_i = 1500$ lbs and $A_i = 12$ in² (for all i). The optimum solution is [160.0, 60.0, 60.0, 8000.0, 8000.0, 6887.9, 7910.1, 8.51, 5.09, 19.72, 3.93, 1.0, 1.01, 2.59, 2.54, 1.0, 1.0] at an objective function value of 11.753. Note that the constraints g_3, g_4, g_5 and g_6 are active at the optimum solution. The constraints fall in the category discussed in Eq. 6.8. Thus the CRSA formulation in Eq. 6.10 can be used. The system parameter σ_{yield} was perturbed from the initial value of 14000 *psi* to observe the trends in the optimum design point. Perturbations of -20% , -15% , -10% , -5% , 5% , 10% , 15% and 20% were applied on σ_{yield} . The optimum design points obtained from the CRSA and linear strategy have been compared.

The percentage change in objective function for the perturbed case with respect to initial case ($\sigma_{yield} = 14000$ *psi*) are illustrated in Fig. 6.6. This percentage change is quantified as $\frac{(f_p^* - f_0^*)}{f_0^*} \times 100\%$, where f_0^* is the optimum objective function value for the initial non-perturbed case and f_p^* is for a $p\%$ perturbation in the system parameter.

The trends observed from this figure illustrate the difference between the linear and the CRSA based post-optimality analysis. As can be clearly seen from Fig. 6.6, the proposed analysis suggests that a better optimum design can be located by changing the yield stress (σ_{yield}) by -10% . This prediction is in agreement with what exact analysis predicts. Comparatively, the linear sensitivity analysis indicates that a change of 20% would correspond to a better optimum design. Thus the linear analysis predicts a wrong optimal candidate. The linear strategy predicts an increase in objective function value for some perturbations, where actually there is

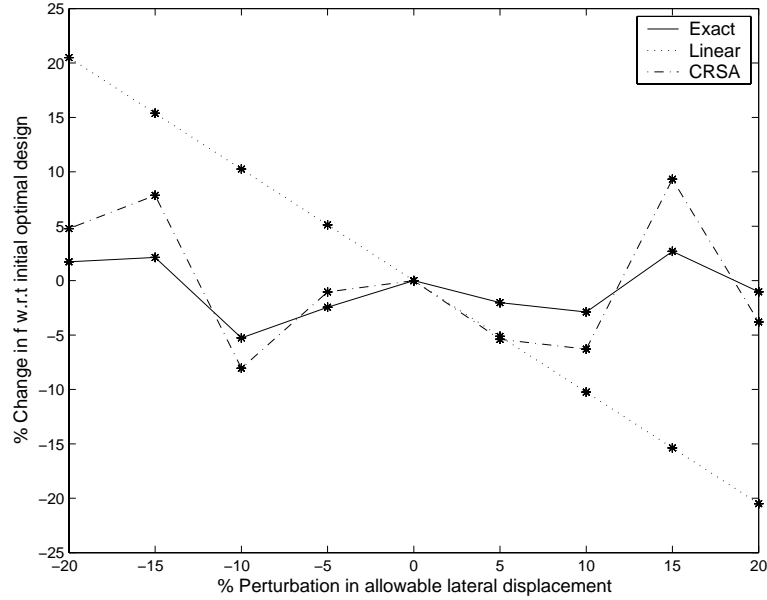


Figure 6.6. Percentage Change in Objective Function with respect to Initial Optimal Design for the HPLC ten-bar problem

a decrease. Hence the results from linear strategy are unreliable as compared to NLPOA method. However it should be noted that there is some error between the CRSA and the exact response of the system. The designer should take this error into account before making any decision.

The percentage error in the objective function of the NLPOA and linear strategy with respect to exact analysis is shown in Fig. 6.7. At any perturbation, the percentage error is quantified as $\frac{(f_a^* - f_e^*)}{f_e^*} \times 100\%$, where f_e^* is the exact optimum objective function value and f_a^* is the function value obtained from linear or CRSA analysis.

The error incurred by the NLPOA strategy is smaller than that experienced in linear analysis. The maximum error of 7% is experienced in NLPOA, however errors

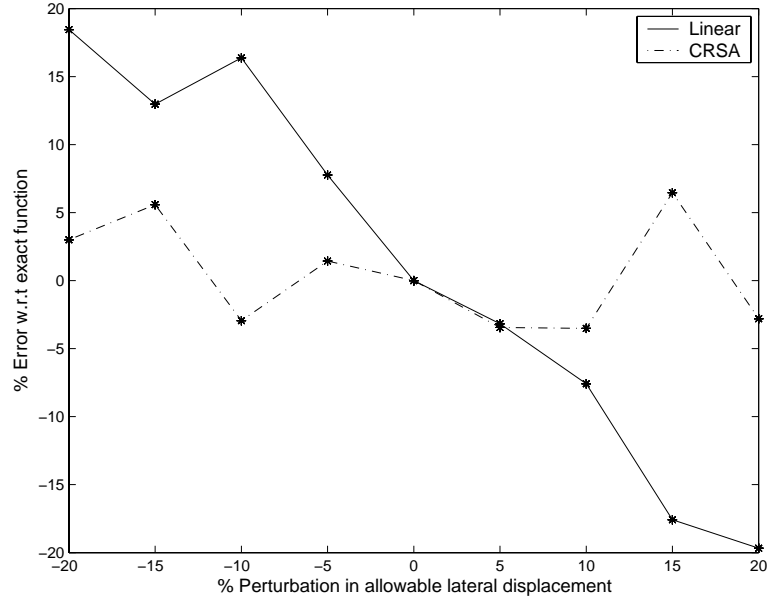


Figure 6.7. Percentage Error with respect to Exact Function for the HPLC ten-bar problem

as high as 20% are encountered in the linear analysis. The linear strategy not only has larger error but also predicts a misleading solution as compared to the NLPOA strategy. At a -10% perturbation in σ_{yield} , the error in NLPOA is close to 3% which is reasonably acceptable. This implies that the post-optimal solution obtained from NLPOA is acceptable within 3% of error. Thus the designer can conclude with good accuracy that a better optimum design is achievable by perturbing the system parameter σ_{yield} by -10% . This kind of analysis will enable the designer to tweak the critical system parameter and locate a better optimum design.

6.6.3 Control-Augmented Structure Problem

The Control-Augmented Structure Problem is described in Sec. A.6. This is an interesting complex MDO problem which requires computationally intensive function calls. The optimum for this problem is [3.0, 3.0, 3.0, 3.0, 3.0, 4.03, 6.99, 9.81, 11.96, 13.71, 0.06] with minimum weight $W = 1493.6$ lbs occurring when 6 design variables are at their bounds. The only active inequality constraint in this problem is g_1 . Here also, the CRSA formulation of constraints in Eq. 6.10 is valid. A similar analysis was done for this problem as was performed on the HPLC problem. In this study, the allowable lateral displacement was perturbed by the same amounts as in the previous test problem. The results are shown in Fig. 6.8 and 6.9.

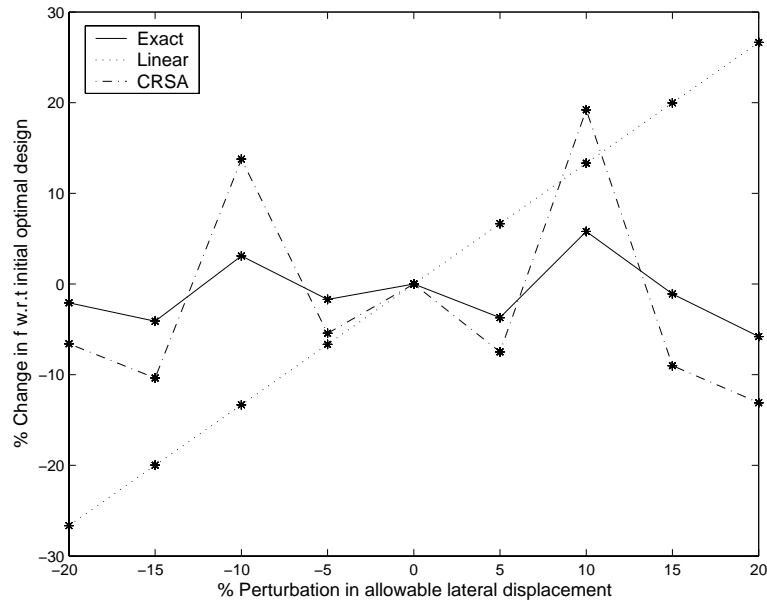


Figure 6.8. Percentage Change in Objective Function with respect to Initial Optimal Design for the Control-Augmented Structure problem

As expected, the linear strategy predicts a linear trend in the change in objective function value. Thus according to the linear strategy, the more the perturbation in system parameter the larger the change in objective function. However the NLPOA method shows that there is no correlation between the percentage perturbation and the change in objective function, which is supported by the exact analysis. The actual trend observed from the NLPOA method is non-linear which is in agreement with the exact analysis. The results shown in Fig. 6.8 illustrate that according to the NLPOA strategy a change of 20% in the lateral displacement will bring the design to a better optimum point. In contrast, the linear strategy predicts that a change of -20% will provide minimum objective function value. Fig. 6.9 shows that at -20% perturbation, the error incurred by linear strategy is close to 25%. The maximum error incurred by the linear strategy is 34%, while the NLPOA only encounter maximum error of 12%. Thus it is not advisable to rely on the solution predicted by the linear strategy. The claim made by NLPOA strategy for achieving a better optimum at 20% perturbation is acceptable within 8% error limits. Thus the NLPOA method provides with more accurate post-optimality studies, which can be used within acceptable error limits.

6.7 Summary

Post-optimality analysis is used to estimate variations in the optimum design when parameter values are changed. Linear post-optimality analysis provides only a first order estimate of the change in the optimum design and objective function

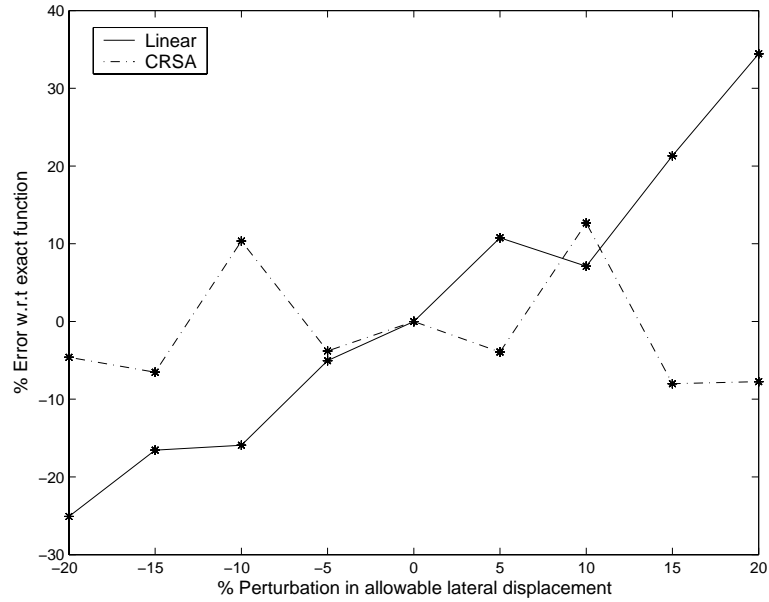


Figure 6.9. Percentage Error with respect to Exact Function for the Control-Augmented Structure problem

value corresponding to small changes in the parameters. Hence a non-linear post-optimality analysis (NLPOA) strategy has been developed. Each local response surface approximation generated during the SAO is archived for use in post-optimality analysis. These local RSAs are blended to give the cumulative response approximation of the system. The NLPOA developed in this research uses CRSAs for the objective function and each constraint. Solution of the CRSA optimization problem gives the new optimum design. This strategy provides an efficient non-linear post-optimality analysis technique which is more accurate than linear estimates. The methodology has been tested on three engineering test problems. The results support the argument that accurate estimates are obtained from the NLPOA.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

This thesis presents the development of an efficient global optimization algorithm along with a description of a non-linear post-optimality analysis. Several implementation issues have been already discussed in Chapters 5 and 6. An overview of conclusions is provided in this chapter along with future recommendations.

7.1 Summary

The major focus of this research is to develop an efficient global optimization strategy, which can be employed by the designer to locate global optimal designs in a reduced time frame. One popular way to ensure fewer system analyses is to construct approximation models. In this study, the global domain is divided into a number of local regions and a Kriging approximation is constructed in each of these regions. A Kriging approximation model was selected because of its capability to characterize the accurate response of non-linear complex systems. However accurate approximations are desired at the same time. Thus one issue this research concentrates on is the development of an accurate approximation scheme. An inexpensive measure of accuracy based on cross-validation is used to validate the approximated

model. The quality of the approximation is improved by adding additional sample points. Gradient based optimizers are employed on each of the approximate models to arrive at the local optimal designs. A multi-start strategy is applied to capture different local optima in a particular local region. The global optimum designs are identified by refining the model fidelity and further analyzing the problem in a reduced trust region around each of the local optimum solutions.

A non-linear post-optimality analysis strategy was also developed in this research. Here also, the exact response of the system was approximated by local response surface approximations. During the sequence of approximate optimization, local approximations are constructed around each optimization iterate. These local approximations are valid within an influence region around each optimization iterate. The cumulative global approximation for post-optimality analysis is obtained by blending the local approximations. The cumulative approximation provides an inexpensive formulation of the system behavior. The post-optimal solution is obtained by performing optimization over the cumulative approximations of the objective function and all the constraints. This strategy provides an inexpensive and non-linear post-optimality analysis. It produces accurate estimates of the post-optimal solution.

7.2 Conclusions

The findings from this research are concluded in this section. Benefits from the developed strategies are briefly described here. The characteristics and features of the strategies are presented along with conclusions for this research study.

7.2.1 GO-LAO Approach

The global optimization strategy using local approximate optimization provides an efficient framework for global optimization. The time saving benefits of the GO-LAO strategy can be used by the designer to arrive at global optimal designs within a reduced time frame. The strategy is devised in such a way that computational time savings are achieved in every possible avenue. A summary of different steps of the GO-LAO algorithm along with advantages is provided below:

- The global domain is divided into several local regions. Kriging approximations are constructed in each of the local regions. This process of model construction can be parallelized in each of the local regions. The approximation in one local region is independent of the other. Thus a Kriging model is constructed in each of the local regions simultaneously. This results in significant time savings.
- Once accurate Kriging approximations are generated in local regions, optimization is performed over each of the local regions. A multi-start approach is used to identify different optima in a particular region. In a particular region, optimization from different starting points is performed in parallel. Not only that, this local multi-start optimization is done simultaneously in all the regions. The time saving benefits are also reaped in this step of the algorithm through parallel computing.
- The local optima obtained from the local optimizations are used to identify the global optimum solutions. The exact function value at each of the local

optima is compared with the predicted value from the Kriging model. The discrepancy between the two values is used to calculate a reduced trust area. The trust region size is calculated as a percentage of the size of the original local region. The fidelity of the Kriging approximation is further improved by sampling in the trust region. Thus accurate approximations are obtained as the algorithm proceeds. Gradient based optimization is again performed in these refined trust regions to provide candidate global optimum solutions.

The parallel computing environment expedites the process of identifying candidate global optimum solutions. Thus the GO-LAO approach provides an efficient global optimization framework. This method facilitates the identification of global optimum solution within a reduced time frame. Thus designers can employ this method to propose new designs before the existing designs becomes obsolete.

7.2.2 NLPOA Strategy

Usually in the post-optimality analysis, a system parameter is modified to observe the trends of the optimal design. A critical parameter in the system is perturbed to observe the change in the optimal design. This kind of analysis demands good accuracy to make a good decision. If the post-optimal solution is not accurate then there is a risk involved in quantifying the new optimum design. The linear post-optimality analysis is based on first order estimates. Hence the estimates from the linear methods may be misleading.

Response surface approximations are constructed at each of the optimization iterates in a sequential approximate optimization process. The local approximation

is valid within a trust region around the optimization iterates. The local RSAs are blended to provide a cumulative approximation. The cumulative response surface approximation provides a good approximation in the neighborhood of local approximated regions by capturing the non-linearities. The CRSA provides an inexpensive formulation of the system behavior. Instead of using the high fidelity and computationally intensive exact analysis, the post-optimal studies can be performed on the explicitly formulated CRSA. This enables a reduction in cost as the inexpensive approximations are used. Thus it gives more reliable results and paves the way for non-linear post-optimality analysis. The implementation of post-optimality analysis by using the NLPOA strategy for a variety of MDO problems demonstrates the potential in obtaining more accurate post-optimal solutions.

7.3 Recommendations for Future Work

It has been shown that global optimization using local approximate optimization provides global optima within a reduced time frame. This study identifies the important characteristics, demonstrates some applications and points out the advantages of the GO-LAO approach. However, a few important technical issues still remain to be examined and developed. The purpose of this section is to highlight these additional issues that are unresolved and worthy of further study.

7.3.1 Division of Global Space

The method of global design space division into local regions is described in Chapter 5. For problems with a small number of design variables, the number of

regions is small and easy to handle. However the number of regions grows with the number of design variables. Hence the space division becomes a crucial step which undermines the time savings of the algorithm. The present approach divides the domain on the basis of model parameters. Initially, the domain is divided into a small number of regions and then sub-divided based on the value of model parameters obtained in that particular region. This step of the algorithm needs more attention in the future. The present approach has the potential for transforming into a more concrete strategy and is open to further study.

7.3.2 Kriging Model Assessment

An approximate cross-validation measure of accuracy to validate the Kriging approximation was discussed in Chapter 2. This is an inexpensive statistic to assess the approximation. There are other statistics available for validating the model. These include Root Mean Square Error, Mean Absolute Error, Cross Validation Root Mean Square Error and others. Unfortunately those are computationally expensive methods. Thus the present statistic is better in terms of computational expense. It is unknown if one statistic is better than the other in terms of quantifying model accuracy. Future studies could focus on examining these statistics for model assessment.

The model validation method assess the ability of the model to predict the expected value or to act as a deterministic model. There has been no assessment to address the uncertainty in the model itself. For a further study, another issue could be understanding the uncertainty in the Kriging model. Note that *uncertainty* is

different from *inaccuracy*. Uncertainty is the inherent error in the model, which is different from inaccurate prediction by the model. This issue remains a subject of interest for the future work.

7.3.3 Improving Accuracy in Cumulative Approximations

The method of improving the accuracy of the cumulative approximation was discussed in Chapter 3. The quality of the CRSA is improved by using additional sample points. This method is currently used in this study. Another method for improving the accuracy was pointed out in Chapter 3. This deals with choosing the appropriate influence size parameter. This method was not explored in the current study.

APPENDIX A

TEST PROBLEMS

A.1 Camelback Six Hump Function

The six hump camelback function is described as [27]:

$$\begin{aligned}
 \text{Minimize.} \quad & f(\mathbf{x}) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + (4x_2^2 - 4)x_2^2 \\
 \text{Subject to:} \quad & -2 \leq x_1 \leq 2 \\
 & -1 \leq x_2 \leq 1
 \end{aligned} \tag{A.1}$$

A.2 Engine Design Problem

This problem deals with engine design and was used in Papalambros and Wagner [38]. The geometry for the compound valve head design is shown in Fig. A.1. The nomenclature for the engine model is tabulated in Table A.1.

The objective in this problem is to maximize the brake power. The mathematical analysis of the model is presented below:

$$\eta_v = \eta_{vb} \frac{1 + 5.96 \times 10^{-3}w^2}{1 + [(9.428 \times 10^{-5}(4v/\pi N_c N_v C_s)(w/d_I^2))]^2} \tag{A.2}$$

$$\eta_{vb} = 1.067 - 0.038 \exp^{w-5.25} \quad \text{for } w \geq 5.25 \tag{A.3}$$

$$= 0.637 + 0.13w - 0.014w^2 + 0.00066w^3 \quad \text{for } w < 5.25$$

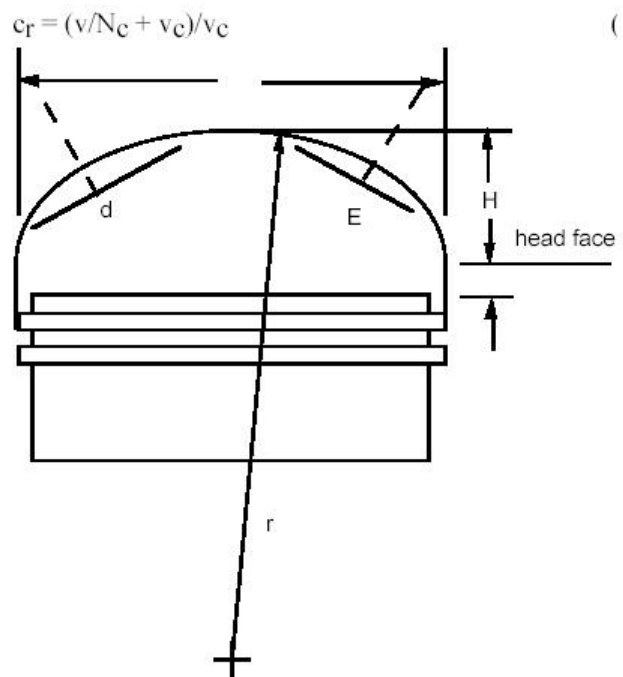


Figure A.1. Schematic for the Compound Valve Head

TABLE A.1

NOMENCLATURE FOR ENGINE MODEL

$FMEP$	Friction mean effective pressure (<i>bars</i>)
A_f	Air/fuel ratio
b	Cylinder bore (<i>mm</i>)
c_r	Compression ratio
C_s	Port discharge coefficient
d_E	Exhaust valve diameter (<i>mm</i>)
d_I	Intake valve diameter (<i>mm</i>)
h	Compound valve chamber deck height (<i>mm</i>)
H	Distance dome penetrates head (<i>mm</i>)
N_c	Number of cylinders
N_v	Number of valves
Q	Lower heating value of fuel (kJ/kg)
r	Radius of compound valve chamber curvature (<i>mm</i>)
s	Stroke of piston (mm)
S_v	Surface to volume ratio (mm_{-1})
v	Displacement volume (mm^3)
v_c	Clearance volume (mm^3)
v_d	Dome volume (mm^3)
V_p	Mean piston speed <i>m/min</i>
w	Revolutions per minute at peak power ($\times 10^{-3}$)
Z_b	RPM factor in volumetric efficiency
Z_n	Mach index of port/chamber design
γ	Ratio of specific heats of in-cylinder gases

$$\eta_{tw} = 0.8595(1 - c_r^{-0.33}) - S_v \quad (\text{A.4})$$

$$\eta_t = \eta_{tad} - S_v \left(\frac{1.5}{w} \right)^{0.5} \quad (\text{A.5})$$

$$\eta_{tad} = 0.9(1 - c_r^{-0.33})(1.18 - 0.225\phi) \quad \text{for } \phi \leq 1 \quad (\text{A.6})$$

$$= 0.9(1 - c_r^{-0.33})(1.655 - 0.7\phi) \quad \text{for } \phi > 1$$

$$S_v = 0.83 \left[8 + 4c_r + 1.5(c_r - 1)\pi b^3 \frac{N_c}{v} \right] / [2 + c_r]b \quad (\text{A.7})$$

$$v_c = \frac{\pi h b^2}{4} + v_d \quad (\text{A.8})$$

$$v_d = \frac{1}{3}\pi \left[(r^2 - b^2/4)^{1.5} - (r^2 - d_I^2/4)^{1.5} - (r^2 - d_E^2/4)^{1.5} \right] - \quad (\text{A.9})$$

$$\pi r^2 \left[(r^2 - b^2/4)^{0.5} - (r^2 - d_I^2/4)^{0.5} - (r^2 - d_E^2/4)^{0.5} \right] - \frac{2}{3}\pi r^3$$

$$H = r - (r^2 - b^2/4)^{0.5} \quad (\text{A.10})$$

$$V_p = \frac{8v}{\pi N_c} w b^{-2} \quad (\text{A.11})$$

$$FMEP = 4.826(c_r - 9.2) + (7.97 + 0.253V_p + 9.7 \times 10^{-6}V_p^2) \quad (\text{A.12})$$

This problem has 8 design variables, 2 equality constraints and 9 inequality constraints. In the standard form, the problem can be stated as:

$$\begin{aligned} \text{Maximize:} \quad & f(\mathbf{x}) = K_0(FMEP - \left(\frac{\rho Q}{A_f}\right)\eta_t\eta_v)wv \\ \text{Subject to:} \quad & h_1 = c_r - \left(\frac{v}{N_c} + v_c\right)/v_c = 0 \\ & h_2 = h - K_{11}b = 0 \\ & g_1 = K_1N_cb - L_1 \leq 0 \\ & g_2 = (4K_2v/\pi N_v L_2)^{0.5} - b \leq 0 \\ & g_3 = (d_I^2 - H^2)^{0.5} + (d_E^2 - H^2)^{0.5} - K_{3c}b \leq 0 \\ & g_4 = K_4d_I - d_E \leq 0 \\ & g_5 = d_E - K_5d_I \leq 0 \end{aligned} \quad (\text{A.13})$$

$$g_6 = 9.428 \times 10^{-5}(4v/\pi N_c)(w/d_I^2) - K_6 C_s \leq 0$$

$$g_7 = c_r - 13.2 + 0.045b \leq 0$$

$$g_8 = w - K_7 \leq 0$$

$$g_9 = 3.6 \times 10^6 - K_8 Q \eta_{tw} \leq 0$$

$$g_{10} = b^2/4 - r^2 \leq 0$$

$$70 \leq b \leq 90$$

$$6 \leq c_r \leq 12$$

$$25 \leq d_E \leq 50$$

$$25 \leq d_I \leq 50$$

$$5 \leq w \leq 12$$

$$1.6 \times 10^6 \leq v \leq 2.3 \times 10^6$$

$$10^{-6} \leq h \leq 10^3$$

$$10^{-6} \leq r \leq 10^3$$

where the values of the design parameters can be found in Table A.2.

A.3 Underwater Autonomous Vehicle Design Problem

The problem involves a stochastic simulation based design tool that allows for optimization to occur in a dynamic environment. The simulation includes a full dynamic model of an Autonomous Underwater Vehicle (AUV). This system was developed by Yukish *et al.* [62] for U.S. Office of Naval Research. It was further developed by Gano *et al.* [15] to evaluate the effectiveness of such a craft given a set of physical attributes. These attributes include but are not limited to: speed,

TABLE A.2

ENGINE DESIGN SPECIFICATION PARAMETERS

Parameter	Specification	Value
A_f	Air/fuel ratio	14.6
C_s	Port discharge coefficient	0.44
N_c	Number of cylinders	4
N_v	Number of valves	2
Q	Lower heating value of fuel	43958 kJ/kg
ρ	Density of inlet charge	1.225 kg/m ³
ϕ	Equivalence ratio	1
K_0	Unit conversion, 4 stroke engine	1/120
K_1	Cylinder separation as % of bore	1.2
K_2	Engine height as multiple of stroke	2
K_{3c}	$1-K_{12}$	0.875
K_4	Lower bound on valve ratio	0.83
K_5	Upper bound on valve ratio	0.89
K_6	Upper bound on Mach Index	0.6
K_7	Upper bound on rpm	6.5
K_8	Upper bound on <i>isfc</i>	230.5 g/kWh
K_{11}	Bore fraction specification for deck height	1/64
K_{12}	Bore fraction specification for valve distance	0.125
L_1	Upper bound on engine block length	400 mm
L_2	Lower bound on engine block length	200 mm

mass, moments of inertia, control gains in the auto-pilot, and target detection capabilities. The effectiveness of the AUV is based upon the probability that it can successfully complete a given mission and how quickly it can complete this mission. The model will ultimately be coupled with the Applied Research Lab's unclassified AUV problem that can compute weights, speeds, and efficiencies based on propulsor types [4], sonar configurations, and various other subsystems. Specifically, this analysis computes most of the inputs needed for the simulation based on the physical components of the vehicle.

In order to use this model in an optimization framework, a mission was selected. This mission was simply to hit an oncoming torpedo before it was able to hit its target. The objective of such a mission is to maximize the probability of successfully hitting the torpedo before the torpedo reaches its own target. In order to calculate this probability, the simulation was run with many different starting configurations including: different speeds of the on coming torpedo, evasive maneuvers of the oncoming torpedo, and also various spacial orientations between the AUV and the enemy torpedo.

Computing the probability that the AUV is successful, requires the model to be run many times and, therefore, is computationally expensive. Also, since the objective function is based at least in part on a discrete outcome, hit or miss, this function is not as smooth as most optimizers require.

A.3.1 Dynamic Model Details

For the objective function, a six-degree of freedom dynamics model of an AUV is used [43]. The model assumes: (1) constant thrust, (2) no currents or disturbances, (3) the center of buoyancy (CB) is located at the center of gravity (CG) and the buoyancy force is always equivalent to the gravitational force, and (4) no lift due to hull. As done in differential game theory [17], this model incorporates an evader and a pursuer. The evader model includes four path scenarios: a straight line path, a barrel roll maneuver, a curved trajectory, and a weaving path. These paths are modeled using simple line and sinusoid equations.

The pursuing AUV is a six-degree of freedom model and includes added mass and cross terms as well as cross-flow drag to make this a nonlinear model. The torpedo uses PID controllers that control its heading in the xy - (rudder) and xz - (elevator) planes. The controller uses the evader position to minimize the angle between the velocity vector of the pursuing torpedo and the direction vector of the target (evading torpedo). The current model is based in the body-fixed frame of the pursuing torpedo; thus the position of the evading torpedo is always relative to the pursuer. A visualization of the AUV in pursuit of the torpedo is shown in Fig. A.2.

A.3.2 Design Problem

The mission chosen for the AUV simulation was to hit an on coming torpedo that is trying to hit the launch platform. The starting point for the AUV is the center

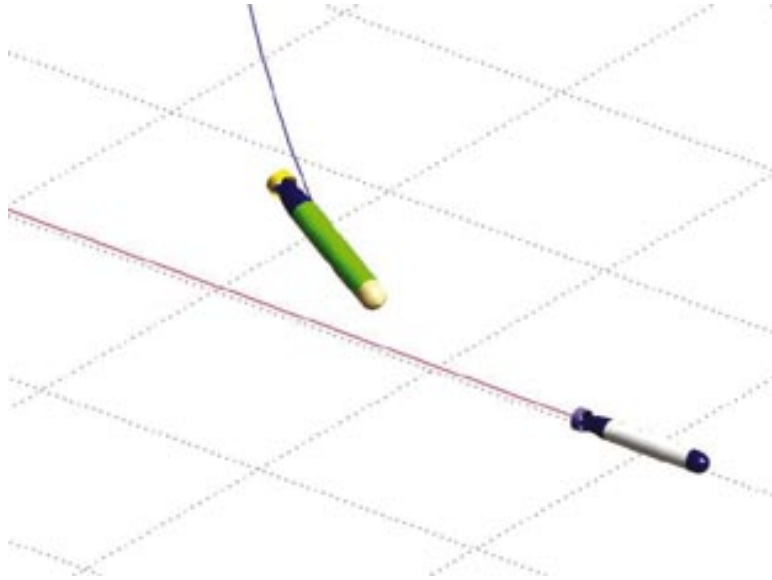


Figure A.2. Visualization of a Simulation in which the AUV is Closing in on the Targeted Torpedo

of the submarine, this represents the submarine launching the AUV in response to an incoming enemy torpedo.

The enemy torpedo will be referred to as the evader because it is trying to evade the AUV and likewise the AUV will be referred to as the pursuer because it is pursuing the torpedo. This may seem confusing at first because in the larger scheme the enemy torpedo is ultimately pursuing the submarine but this simulation is focused only on the AUV and the torpedo, therefore in that reference frame the AUV is the pursuing vehicle.

For each simulation used in this example the evader is started from each of 5 different yaw positions relative to the submarine located on a 100 meter semi-circle as seen in Fig. A.3. The yaw angles used here are also shown in Fig. A.3. From

each of these yaw positions three different starting height of the torpedo are used one is level with the submarine and the other two are 10 meters above and below submarine. Two more variations are used for each of these spacial starting points and those are the evading maneuvers used by the torpedo and also the speed at which the torpedo can travel. Four different evading maneuvers were used: straight path to the submarine, barrel rolling path, a half-sinusoidal path, and a two dimensional weave in the plane of the submarine. The evading torpedo was also given two speeds which were held constant for a single simulation these speeds were 10 m/s and 25 m/s . The 5 starting yaws, 3 heights, 4 paths, and 2 different speeds result in 120 different simulation runs. For each simulation if the AUV comes within a radius of twice its length to the torpedo it is considered a kill or a success. If the torpedo evades the AUV and hits the submarine the run is considered a failure. In either case the total simulation time to either hit or miss and also the closest encounter distance are calculated.

The design problem is to maximize a combination of the mission success rate and average time to kill. Design variables include sonar sizing, propellant mass, thrust available, payload size, and various control gains used in the differential games based guidance system.

A.4 Barnes Problem

This is a purely mathematical problem and was originally formulated by G. K. Barnes as part of his Master's Thesis [2]. The problem is stated as:

$$\text{Minimize:} \quad f(\mathbf{x}, \mathbf{y}) = a_1 + a_2x_1 + a_3y_4 + a_4y_4x_1 + a_5y_4^2$$

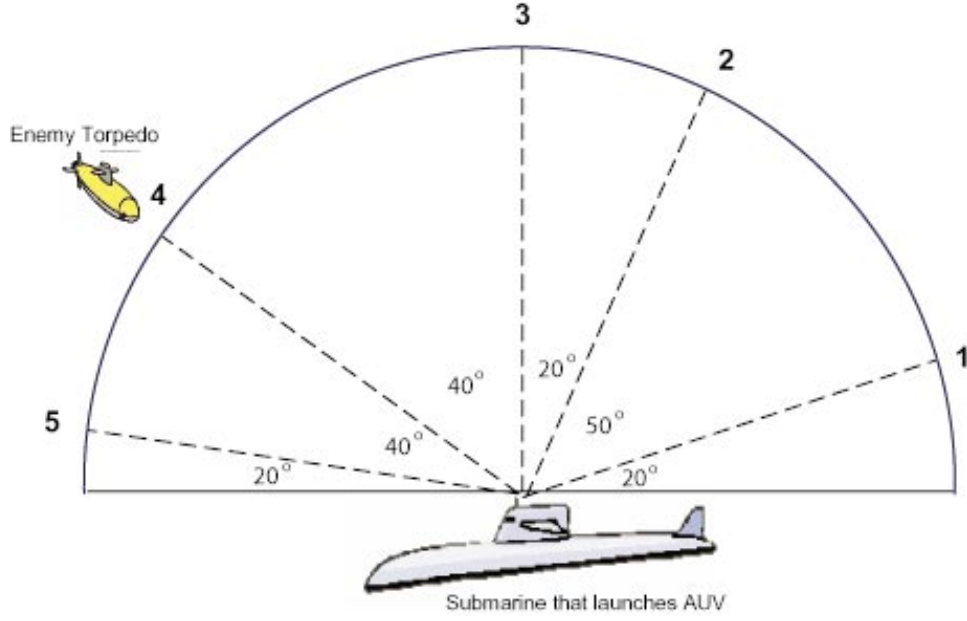


Figure A.3. AUV Mission Scenario

$$\begin{aligned}
 &+ a_6x_2 + a_7y_1 + a_8x_1y_1 + a_9y_1y_4 + a_{10}y_2y_4 \\
 &+ a_{11}y_3 + a_{12}x_2y_3 + a_{13}y_3^2 + \frac{a_{14}}{x_2 + 1} + \\
 &a_{15}y_3y_4 + a_{16}y_1y_4x_2 + a_{17}y_1y_3y_4 + a_{18}x_1y_3 \\
 &+ a_{19}y_1y_3 + a_{20}e^{a_{21}y_1}
 \end{aligned}$$

$$\text{Subject to: } g_1 = 1 - \frac{y_1}{700} \leq 0 \tag{A.14}$$

$$g_2 = \frac{y_4}{(25)^2} - \frac{x_2}{5} \leq 0$$

$$g_3 = \left(\frac{x_1}{500} - 0.11\right) - (y_5 - 1)^2 \leq 0$$

$$0.0 \leq x_1 \leq 75.0$$

$$0.0 \leq x_2 \leq 75.0$$

where the coefficients a are constants and their values can be found in Table. A.3.

The states are calculated by the Contributing Analyses CAs as:

$$CA_1 \quad y_1 = x_1 x_2$$

$$y_3 = x_2^2$$

$$CA_2 \quad y_2 = y_1 x_1$$

$$y_4 = x_1^2$$

$$y_5 = \frac{x_2}{50}$$

A.5 High Performance Low Cost (HPLC) Structure Problem

This problem is posed as the optimal design of a high performance low cost (HPLC) structural system as shown in Fig. A.4. A detailed description of the problem is given in Wujek [59]. The design variables in this problem come from three different disciplines. The configuration of the structure is varied in order to explore different topologies. Thus, the length of the rectangular first bay (L_1) and the top and bottom lengths of the outer bay (L_2, L_3) are selected as geometric design variables. The masses placed on all of the unconstrained nodes ($M_1 - M_4$) are structural design variables representing the payload. The areas of the truss members ($A_1 - A_{10}$) make up the final category of design variables since sizing is one of the main design considerations. Thus the problem consist of 17 design variables.

The objective is to find the size and shape of the truss such that the weight (W_{tot}) of the structure is a minimum (low cost) and the loads (P_i) it is capable of sustaining & the payload (M_i) it carries are a maximum (high performance). This

TABLE A.3

COEFFICIENTS FOR THE BARNES PROBLEM

a_1	75.196	a_2	-3.8112
a_3	0.12694	a_4	-2.0567e-3
a_5	1.0345e-5	a_6	-6.8306
a_7	0.030234	a_8	-1.28134e-3
a_9	3.5256e-5	a_{10}	-2.266e-7
a_{11}	0.25645	a_{12}	-3.4604e-3
a_{13}	1.3514e-5	a_{14}	-28.106
a_{15}	-5.2375e-6	a_{16}	-6.3e-8
a_{17}	7.0e-10	a_{18}	3.4054e-4
a_{19}	-1.6638e-6	a_{20}	-2.8673
a_{21}	0.0005		

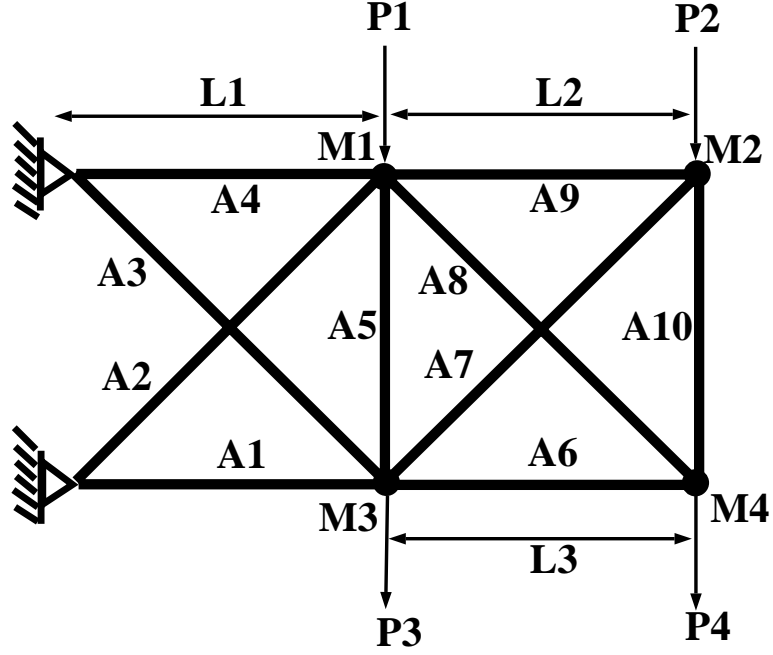


Figure A.4. Schematic of the HPLC Structure

multi-objective problem can be formulated in a single objective problem by defining a cost-performance index (*CPI*) which includes each of the objectives. The design is subject to minimum payload and load requirements as well as yield stress and first natural frequency constraints. A total of 13 inequality constraints are defined for this problem. In standard form the system optimization problem is:

$$\begin{aligned}
 \text{Minimize:} \quad & f(\mathbf{x}) = CPI = w_1 W_{tot} + \frac{w_2}{\sum P_i} + \frac{w_3}{\sum M_i} \\
 \text{Subject to:} \quad & g_1 = 1.0 - \frac{\sum M_i}{(M_{tot})_{min}} \leq 0.0 \\
 & g_2 = 1.0 - \frac{\sum P_i}{(P_{tot})_{min}} \leq 0.0 \\
 & g_3 = 1.0 - \frac{\omega_1}{\omega_{1,min}} \leq 0.0
 \end{aligned} \tag{A.15}$$

$$g_{4-13} = 1.0 - \frac{\sigma_{yield}}{|\sigma_{1-10}|} \leq 0.0$$

$$160 \leq x_1 \leq 500$$

$$60 \leq x_{2-3} \leq 500$$

$$250 \leq x_{4-7} \leq 8000$$

$$1 \leq x_{8-17} \leq 20$$

$$where : \quad w_1 = .003, \quad w_2 = 10^6, \quad w_3 = 3.5 \times 10^6$$

$$(M_{tot})_{min} = 5000 \text{ lbs}, \quad (P_{tot})_{min} = 100,000 \text{ lbs}$$

$$\omega_{1,min} = 2.0 \text{ Hz}, \quad \sigma_{yield} = 14000 \text{ psi}$$

The coefficients w_i in the objective function are introduced to weight the separate components so that none of the component dominate others in driving the optimization. The yield stress of 14000 *psi* is based on the choice of aluminum as the material for the structure. The loads (P_i) applied to the structure are defined to be a function of the lengths of the bays (L_i) and the payload masses (M_i) placed on the structure as shown in eq. A.16.

$$P_i = \sum_{k=1}^3 a_k^i \left(\frac{L_k}{L_{ref}} \right)^{b_k^i} + \sum_{j=1}^4 c_j^i \left(\frac{M_j}{M_{ref}} \right)^{d_j^i} \quad (\text{A.16})$$

This is similar to an aero-elastic structure in which the loads incurred are dependent on the size and shape of the structure. The coefficients (a , b , c and d) in eq. A.16 are chosen so as to apply greater emphasis to the effect that certain lengths or masses have on the given loads.

A.6 Controls-Augmented Structure Problem

The controls-augmented structure design problem shown in fig. A.5 was introduced by Sobieszczanski-Sobieski *et al.* [56]. The problem comprises a total of 11 design variables and 43 states. The physical problem consists of a cantilever beam subjected to static loads along the beam and to a dynamic excitation force applied at the free end. Two sets of actuators are placed at the free end of the beam to control both the lateral and rotational displacement.

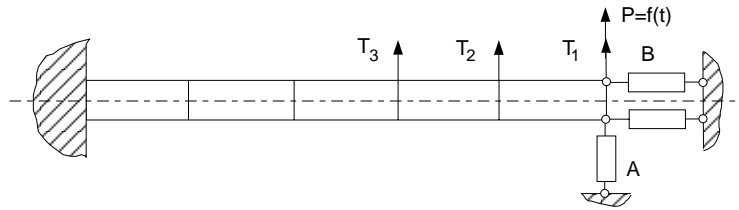


Figure A.5. Controls Augmented Structure

The system analysis is comprised of two coupled contributing analysis as shown in Fig. A.6. The structures subsystem, CA_s consists of a finite element model of the beam where the natural frequencies and modes of the cantilever beam are computed. CA_s requires the characteristics of the beam and the weight of the controls system as input. The weight of the controls system is calculated at the controls CA, CA_c . The weight of the controls system is a function of the dynamic displacements and rotations of the free end of the beam. These dynamic displacements and rotations are functions of the natural frequencies and modes obtained in the structures CA, thus subjecting these CA's to coupling.

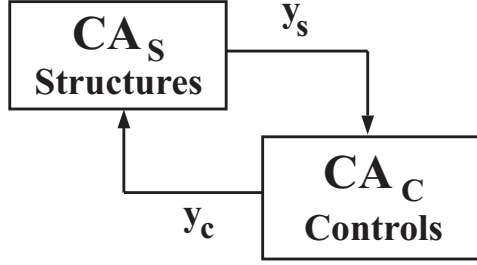


Figure A.6. Coupled System of the Controls-Augmented Structure problem

The objective of the optimization is to minimize the total weight of the system W_t , composed of the weight of the beam W_s plus the weight of the control system W_c . The minimization is subjected to seven constraints on the static stresses, lateral and rotational displacements, natural frequencies and dynamic lateral and rotational displacements at the free end of the beam. The problem is posed as:

$$\begin{aligned}
 \text{Minimize:} \quad & W_t = W_s + W_c \\
 \text{Subject to:} \quad & g_1 = \frac{dl}{dl_a} - 1 \leq 0 \\
 & g_2 = \frac{dr}{dr_a} - 1 \leq 0 \\
 & g_3 = 1 - \frac{\omega_1}{\omega_{1a}} \leq 0 \\
 & g_4 = 1 - \frac{\omega_2}{\omega_{2a}} \leq 0 \\
 & g_5 = \frac{\sigma}{\sigma_a} - 1 \leq 0 \\
 & g_6 = \frac{ddl}{ddl_a} - 1 \leq 0 \\
 & g_7 = \frac{ddr}{ddr_a} - 1 \leq 0 \\
 & 3.0 \leq x_{1-10} \leq 36.0 \\
 & 0.01 \leq x_{11} \leq 0.06
 \end{aligned} \tag{A.17}$$

where dl is the static lateral displacement, dr is the static rotational displacement, ddl is the dynamic lateral displacement, ddr is the dynamic rotational displacement, ω_1 is the first natural frequency, ω_2 is the second natural frequency, and σ is the static stress. The subscript a stands for the allowed value.

BIBLIOGRAPHY

- [1] Baker, J. E. 1985: Adaptive selection methods for genetic algorithms. In *Proceedings of 1st International Conference on Genetic Algorithms and their Applications*, pp. 370–374. Lawrence Erlbaum Associates, Hillsdale, NJ.
- [2] Barnes, G. K. 1967: Master’s thesis, The University of Texas, Austin, Texas.
- [3] Belegundu, A. D., Chandrupatla, T. R. 1999: *Optimization Concepts and Applications in Engineering*. Prentice Hall.
- [4] Breslin, J. P., Anderson, P. 1994: *Hydrodynamics of Ship Propellers*. Cambridge University Press.
- [5] Chandila, P. K. 2002: *Adaptive Slicing of Free Form Surfaces using Ruled Surfaces with Optimum Contour Points*. Bachelor’s Thesis, Indian Institute of Technology, Kharagpur, India.
- [6] Chandila, P. K., Agarwal, H., Renaud, J. E. 2004: An efficient strategy for global optimization using local kriging approximations. In *Proceedings of the 45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, AIAA 2004-1873. Palm Springs, CA. 19 - 22 April.
- [7] Chandila, P. K., Pérez, V. M., Renaud, J. E. 2004: Post-optimality analysis using a cumulative global response surface approximation. In *Proceedings of the 42nd AIAA Aerospace Sciences Meeting and Exhibit*, AIAA 2004-144. Reno, Nevada. 5 - 8 January.
- [8] Choi, B. K. 1990: *Modelling of Curves and Surfaces for Computer Aided Design*. McGraw-Hill.
- [9] Davis, L. 1991: *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York, NY.
- [10] Fang, K. T., Lin, D. K. J., Winker, P., Zhang, Y. 2000: Uniform design: Theory and application. *Technometrics*, volume 42, no. 2, pp. 237–248.
- [11] Farin, G. 1990: *Curves and Surfaces for Computer Aided Geometrical Design*. Academic Press, New York, 2nd edition.
- [12] Floudas, C. A., Pardalos, P. M. 1990: *A collection of Test Problems for Constrained Global Optimization Algorithms*. Lecture Notes in Computer Science, Springer-Verlag.

- [13] Fogarty, T. C. 1989: Varying the probability of mutation in the genetic algorithm. In *Proceedings of 3rd International Conference on Genetic Algorithms and their Applications*, pp. 104–109. George Mason University.
- [14] Gano, S. E. 2003: Sensitivity of optimal conforming airfoils to exterior shape. In *Proceedings of the AIAA Student Conference 2003*. University of Kentucky at Paducah. April 4-5.
- [15] Gano, S. E., Patel, N. M., Renaud, J. E., Martin, J. D., Yukish, M. 2004: Simulation based optimization of an autonomous underwater vehicle. In *Proceedings of the 45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, AIAA-2004-2034. Palm Springs, CA. 19 - 22 April.
- [16] Houck, C. R., Joines, J. A., Kay, M. J. 1995: A genetic algorithm for function optimization: A matlab implementation. Technical Report 95-09, NCSU-IE TR.
- [17] Isaacs, R. 1965: *Differential Games*. Dover Edition.
- [18] Jin, R., Chen, W., Simpson, T. W. 2001: Comparative studies of metamodelling techniques under multiple modelling criteria. *Structural Multidisciplinary Optimization*, volume 23, pp. 1–13.
- [19] Jones, D. R. 2001: A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, volume 21, pp. 345–383.
- [20] Jones, D. R., Schonlau, M., Welch, W. J. 1998: Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, volume 13, no. 4, pp. 455–492.
- [21] Khuri, A. I., Cornell, J. A. 1996: *Response Surface: Designs and Analyses*. Statistics - Textbooks and Monographs, Vol. 52, Marcel Dekker Inc.
- [22] Laskari, E. C., Parsopoulos, K. E., Vrahatis, M. N. 2003: Evolutionary operators in global optimization with dynamic search trajectories. *Numerical Algorithms*, volume 34, pp. 393–403.
- [23] Liu, P., Hartzell, S., Stephenson, W. 1995: Nonlinear multiparameter inversion using a hybrid global search algorithm - applications in reflection seismology. *Geophysical Journal International*, volume 122, pp. 991–1000.
- [24] Lophaven, S. N., Nielsen, H. B., Søndergaard, J. 2002: Aspects of the matlab toolbox dace. Technical Report IMM-REP-2002-13, Technical University of Denmark, DK-2800 Kongens Lyngby - Denmark.
- [25] Lophaven, S. N., Nielsen, H. B., Søndergaard, J. 2002: Dace a matlab kriging toolbox. Technical Report IMM-REP-2002-12, Technical University of Denmark, DK-2800 Kongens Lyngby - Denmark. Version 2.0.
- [26] Martin, J. D. 2003: *A Methodology for Evaluating System-Level Uncertainty in the Conceptual Design of Complex Multidisciplinary Systems*. Ph.D. Thesis Proposal, Pennsylvania State University, Pennsylvania.

- [27] Martin, J. D., Simpson, T. W. 2003: A study on the use of kriging models to approximate deterministic computer models. In *Proceedings of ASME 2003 Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, DET2003/DAC-48762, pp. 1–10. Chicago, IL. 2-6 September.
- [28] Matheron, G. 1963: Principles of geostatistics. *Economic Geology*, volume 58, pp. 1246–1266.
- [29] McAllister, C. D., Simpson, T. W. 2003: Multidisciplinary robust design optimization of an internal combustion engine. *Transactions of the ASME*, volume 125, pp. 124–130.
- [30] Meckesheimer, M., Barton, R. R., Limayem, F., Yannou, B. 2000: Metamodeling of combined discrete/continuous responses. In *Design Theory and Methodology - DTM'00*, ASME. Baltimore, MD.
- [31] Meckesheimer, M., Bokker, A. J., Barton, R. R., Simpson, T. W. 2002: Computationally inexpensive metamodel assessment strategies. *AIAA Journal*, volume 40, no. 10, pp. 2053–2060.
- [32] Michalewicz, Z. 1992: *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, New York, NY.
- [33] Mitchell, T. J., Morris, M. D. 1992: Bayesian design and analysis of computer experiments: Two examples. *Statistica Sinica*, volume 2, pp. 352–379.
- [34] Mortenson, M. E. 1985: *Geometric Modeling*. John Wiley and Sons, New York.
- [35] Myers, R. H., Montgomery, D. C. 1995: *Response Surface Methodology*. Wiley Series in Probability and Statistics.
- [36] Oh, S. H., Lee, B. C. 2003: Development of an efficient algorithm for global optimization by simplex elimination. *Engineering Optimization*, volume 35, no. 6, pp. 607–625.
- [37] Osio, I. G., Amon, C. H. 1996: An engineering design methodology with multistage bayesian surrogate and optimal sampling. *Research in Engineering Design*, volume 8, no. 4, pp. 189–206.
- [38] Papalambros, P. Y., Wagner, T. C. 1991: Optimal engine design using nonlinear programming and the engine assessment model. Technical report, Ford Motor Company Scientific Research Laboratories and University of Michigan Department of Mechanical Engineering.
- [39] Pérez, V. M., Apker, T. B., Renaud, J. E. 2002: Parallel processing in sequential approximate optimization. In *Proceedings of the 43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, AIAA 2002-1589. Denver, CO. April 22-25.

- [40] Pérez, V. M., Renaud, J. E., Watson, L. T. 2002: An interior point sequential approximate optimization methodology. In *Proceedings of the 10th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA 2002-5505. Atlanta, GA. September 4-6.
- [41] Pérez, V. M., Renaud, J. E., Watson, L. T. 2003: Homotopy curve tracking in approximate interior point optimization. In *Proceedings of the 44th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, AIAA 2003-1670. Norfolk, VA. April 7-10.
- [42] Pham, D. T., Karaboga, D. 2000: *Intelligent Optimization Techniques - Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks*. Springer-Verlag, London.
- [43] Presterio, T. 2001: *Verification of a Six-Degree of Freedom Simulation Model for the REMUS Autonomous Underwater Vehicle*. Master's thesis, University of California at Davis.
- [44] Rasmussen, J. 1998: Nonlinear programming by cumulative approximation refinement. *Structural Optimization*, volume 15, pp. 1-7.
- [45] Renaud, J. E. 1989: *Using the Generalized Reduced Gradient Method to Handle Equality Constraints directly in a Multilevel Optimization*. Master's thesis, Rensselaer Polytechnic Institute, Troy, New York.
- [46] Rodríguez, J. F., Pérez, V. M., Padmanabhan, D., Renaud, J. E. 2001: Sequential approximate optimization using variable fidelity response surface approximations. *Structural and Multidisciplinary Optimization*, volume 22, pp. 24-34. Published by Springer-Verlag.
- [47] Rodríguez, J. F., Renaud, J. E., Watson, L. T. 1998: Trust region augmented lagrangian methods for sequential response surface approximation and optimization. *AIAA Journal*, volume 32, no. 1, pp. 198-205.
- [48] Sacks, J., Schiller, S. B., Welch, W. J. 1989: Design for computer experiments. *Technometrics*, volume 31, no. 1, pp. 41-47.
- [49] Sacks, J., Welch, T. J., W. J. Mitchell, Wynn, H. P. 1989: Design and analysis of computer experiments. *Statistical Science*, volume 4, no. 4, pp. 409-435.
- [50] Sasena, M. J., Papalambros, P., Goovaerts, P. 2002: Exploration of meta-modeling sampling criteria for constrained global optimization. *Engineering Optimization*, volume 34, pp. 263-278.
- [51] Schaffer, J. D., Caruana, R. A., Eshelman, L. J., Das, R. 1989: A study of control parameters affecting on-line performance of genetic algorithms for function optimisation. In *Proceedings of 3rd International Conference on Genetic Algorithms and their Applications*, pp. 51-61. George Mason University.
- [52] Schonlau, M. 1997: *Computer Experiments and Global Optimization*. Ph.D. thesis, University of Waterloo, Ontario, Canada.

- [53] Simpson, T. W., Mauery, T. M., Korte, J. J., Mistree, F. 1998: Comparison of response surface and kriging models for multidisciplinary design optimization. In *Proceedings of 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA-98-4755, pp. 381–391. St. Louis, MO.
- [54] Simpson, T. W., Peplinski, J. D., Koch, P. N., Allen, J. K. 2001: Metamodels for computer-based engineering design: Survey and recommendations. *Engineering with Computers*, volume 17, pp. 129–150.
- [55] Sobieszczanski-Sobieski, J., Barthelemy, J. F., Riley, K. M. 1982: Sensitivity of optimum solutions to problem parameters. *AIAA Journal*, volume 20, no. Sept., pp. 1291–1299.
- [56] Sobieszczanski-Sobieski, J., Bloebaum, C. L., Hajela, P. 1991: Sensitivity of control-augmented structure obtained by a system decomposition method. *AIAA Journal*, volume 29, no. 2, pp. 264–270.
- [57] Whitely, D., Hanson, T. 1989: Optimising neural networks using faster, more accurate genetic search. In *Proceedings of 3rd International Conference on Genetic Algorithms and their Applications*, pp. 370–374. George Mason University.
- [58] Wu, W. H., Lin, C. Y. 2004: Hybrid-coded crossover for binary-coded genetic algorithms in constrained optimization. *Engineering Optimization*, volume 36, no. 1, pp. 101–122.
- [59] Wujek, B. A. 1997: *Automation Enhancements in Multidisciplinary Design Optimization*. Ph.D. thesis, University of Notre Dame, Indiana.
- [60] Xu, P. L. 2002: A hybrid global optimization method: The one-dimensional case. *Journal of Computational and Applied Mathematics*, volume 147, pp. 301–314.
- [61] Xu, P. L. 2003: A hybrid global optimization method: The multidimensional case. *Journal of Computational and Applied Mathematics*, volume 155, pp. 423–446.
- [62] Yukish, M., Bennett, L., Simpson, T. W. 2000: Requirements on mdo imposed by the undersea vehicle conceptual design problem. In *Proceedings of the 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA-2000-4816. Long Beach, CA. 6-8 September.
- [63] Zalzal, A. M. S., Fleming, P. J. 1997: *Genetic Algorithms in Engineering Systems*. Control Engineering Series 55, The Institution of Electrical Engineers, London, United Kingdom.