ABSTRACT

Title of Dissertation Online and Offline Approximations for Population Based Multi-Objective Optimization

Genzi Li, Doctor of Philosophy, 2007

Dissertation directed by: Professor Shapour Azarm
Department of Mechanical Engineering
University of Maryland

The high computational cost of population based optimization methods has been preventing applications of these methods to realistic engineering design problems. The main challenge is to devise approaches that can significantly reduce the number of function (or simulation) calls required in such optimization methods. This dissertation presents some new online and offline approximation approaches for design optimization. In particular, it presents new DOE and metamodeling techniques for Genetic Algorithm (GA) based multi-objective optimization methods along four research thrusts. The first research thrust is called: Online Metamodeling Assisted Fitness Evaluation. In this thrust, a new online metamodeling assisted fitness evaluation approach is developed that aims at significantly reducing the number of function calls in each generation of a Multi-Objective Genetic Algorithm (MOGA) for design optimization. The second research thrust is called: DOE in Online Metamodeling. This research thrust introduces a new DOE method that aims at reducing the number of generations in a MOGA. It is shown that the method developed under the second research thrust can, compared to the method in the first thrust, further reduce the number of function calls in the MOGA. The third research thrust is called: DOE in Offline Metamodeling. In this thrust, a new DOE method is presented for sampling points in the non-smooth regions of a design space in order to improve the accuracy of a metamodel. The method under the third thrust is useful

in approximation assisted optimization when the number of available function calls is limited. Finally, the fourth research thrust is called: Dependent Metamodeling for Multi-Response Simulations. This research thrust presents a new metamodeling technique for an engineering simulation that has multiple responses.

Numerous numerical and engineering examples are used to demonstrate the applicability and performance of the proposed online and offline approximation techniques. In particular, it is shown that in situations where the application of population based optimization techniques requires numerous simulation evaluations (or function calls), the proposed online metamodeling assisted fitness evaluation approach, the DOE assisted online metamodeling approach, and the DOE assisted offline metamodeling approach can be employed to construct a global approximation to the simulation model and significantly reduce the number of function calls. Moreover, for simulations with multiple responses, the proposed dependent metamodeling approach can be used to construct reasonably accurate metamodels and thus facilitate optimization.

ONLINE AND OFFLINE APPROXIMATIONS
FOR POPULATION BASED MULTI-OBJECTIVE OPTIMIZATION

By
Genzi Li

Dissertation Submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2007

Advisory Committee:

Dr. Shapour Azarm, Chair/Advisor
Dr. Roberto Celi (Dean’s Representative)
Dr. P. K. Kannan
Dr. Ali Mosleh
Dr. Miao Yu

To My Wife, Dr. Xuefeng Fu

ACKNOWLEDGEMENTS

I wish to express my sincere gratitude to Professor Shapour Azarm for his suggestions, guidance, encouragement, and friendship during my stay at the University of Maryland, College Park. I truly believe that I am indeed lucky to have had him as my advisor, leading me through his marvelous kingdom of engineering design optimization, training me the ability of communications, and demonstrating to me the importance of details.

I would like to acknowledge my committee members: Professor Roberto Celi, Associate Professor P. K. Kannan, Professor Ali Mosleh, and Assistant Professor Miao Yu for accepting the invitation to serve on my thesis committee and for their priceless time, effort, comments, and suggestions for improving the thesis.

I would also like to acknowledge the financial support for my thesis related research that was provided in part through the National Science Foundation Grant DMI 0112767, in part through the Office of Naval Research under contract number N00014-05-1-0529, and in part through the Petroleum Institute of Abu Dhabi, United Arab Emirates, as part of the Education and Energy Research Collaboration agreement between the PI and University of Maryland, College Park. Such support does not constitute an endorsement by these funding agencies of the opinions expressed in the thesis.

During my stay at Design Decision Support Laboratory, I received help and advice on numerous occasions from my colleagues, both present and former. I would like to thank Dr. Ali Farhang Mehr, for guiding me to the fabulous realm of approximation. I was very delighted to work with him. Mr. Mian Li is gratefully acknowledged for providing gracious assistance in developing the methods presented in Chapter 3 and Chapter 4 and

co-authoring joint papers along those lines. Additionally, I would like to thank Dr. Subroto Gunawan, Dr. Kumar Maddulapalli, Dr. Babak Besharati, Mr. Vikrant Aute, and Lt. Nathan Williams, for helping to create an enjoyable environment in our Design Decision Support Laboratory.

Mr. Paul Higgins is gratefully acknowledged for providing gracious assistance with editing of my dissertation.

Special thanks are given to my wonderful wife, Dr. Xuefeng Fu, for her love, devotion, support, encouragement, sharing the ups and downs in the last 18 years, for giving me a joyful life with happiness, excitement, with our gorgeous baby girl, Amy.

Additionally, I would like to thank my parents-in-law, for sharing a very busy but important period of time with me, for helping Xuefeng and me taking care of our daughter, Amy, and for always being ready to support us.

Finally, for providing me the guidance, support, encouragement, and requisite strength and perseverance, to my parents and my sisters I am most thankful.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

B	Flow rate of bottom product
$Corr$	Correlation between estimated responses
Cov	Covariance matrix of true responses
d	Center distance
D	Flow rate of distillate product
$\boldsymbol{D}$	A DOE design
det	Determinant
DOE	Design of Experiments
DoI	Degree-of-Influence
$D(t)$	Maximum displacement
exp	Exponential
$e_{AC}^{x_0}$	Accumulative error for an unobserved point
$e_{LOO}^{x_i}$	LOO error for an experiment
F	Feed flow rate
$\boldsymbol{F}$	Applied force
f_i	Objective function
$\hat{f}_m(.)$	Independent metamodel
$F(t)$	Maximum force
g_j	Inequality constraint
$\hat{g}_m(.)$	Dependent metamodel
GA	Genetic Algorithm
H	Entropy
h_k	Equality constraint
$I_P(x_0)$	Prediction interval
J	Number of inequality constraint
$\boldsymbol{J}$	Column vector of ones
K	Number of equality constraint
L	Liquid molar flow rate
L_c	Crank length
L_r	Rod length
LOO	Leave-One-Out
M	Number of objective functions
MAE	Maximum absolute error
MMD	Minimum of minimum distance
MOGA	Multi-Objective Genetic Algorithm
O	Worst case complexity analysis, and is for “order of magnitude”
P	Feed pressure

Q	Heat generation rate
$\boldsymbol{r}$	Correlation vector for true responses and the estimated response for an unobserved point
R	Reflux ratio
$\boldsymbol{R}$	Correlation matrix of true responses
RMSE	Root mean squared error
$s^2(x_0)$	Kriging variance for an unobserved point x_0
STD	Standard deviation
T	Feed temperature
t_K	Computational time for building kriging metamodels
t_O	Computational time for other inner computation
t_S	Computational time for sampling
U	A set of unobserved points
V	Vapor molar flow rate
V_B	Boil up ratio
V_{in}	Inlet air velocity
x	One dimensional design variable
$\boldsymbol{x}$	Design variable vector
X_B	Composition of the more volatile component in bottom stream
X_N	Mole fraction of more volatile component in liquid leaving tray N
X_D	Composition of the more volatile component in distillate stream
X_F	Mole fraction of the more volatile component in the feed
y	True response
Y	Estimated response
Y_N	Mole fraction of more volatile component in vapor leaving tray N
Z	Stochastic process in kriging
α	DOI diminish factor
δ_j	Consistency measure
θ	Correlation parameter
μ	Mean of the responses based on the current metamodel
$\hat{\mu}$	Estimations for μ
σ^2	Variance for Z
$\hat{\sigma}^2$	Estimations for σ^2
ε	Tolerance parameter used in dependent metamodeling approach
Ω	Input Space
$\boldsymbol{\Omega}^{n+m}$	A combined space of input and response spaces
Λ	Response space

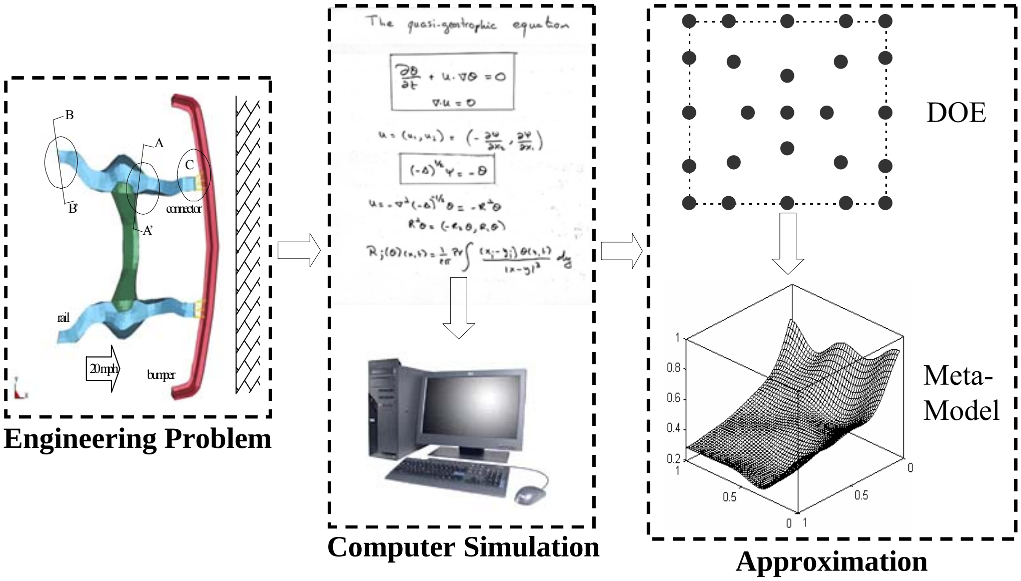
Chapter 1 Introduction

This dissertation presents some new approximation methods. Approximation methods are often used in computational engineering design and optimization where it is necessary to perform design calculations efficiently with minimum possible number of simulation evaluations (or function calls). An approximation method involves two steps: Design of Experiments (DOE) and metamodeling. This dissertation presents a new online metamodeling approach to significantly reduce the number of function calls for population based multi-objective optimization methods, in general, and Multi-Objective Genetic Algorithms (MOGAs), in particular. In addition, a novel DOE method is developed and integrated with the online metamodeling approach to further reduce the number of function calls. Also, a new adaptive DOE approach is developed for offline metamodeling. Finally, a new metamodeling approach is presented for multi-response deterministic simulations. Numerous numerical and engineering examples are used to demonstrate the merits and applicability of the proposed DOE and metamodeling approaches.

1.1 Motivation and Objective

Mathematical models have been widely used by researchers to analyze engineering problems. These mathematical models are often implemented as deterministic computer simulation or analysis models. Engineering design and optimization relies on exploration of different points and therefore numerous analyses or simulation calls are unavoidable. In fact, computational design optimization methods, in particular population based optimization methods, often require numerous simulation runs (or function calls) before the assessment and selection of a particular optimum engineering design alternative is complete.

Today the most common strategy to reduce the number of function calls associated with such an exploration is to use approximation (see, e.g., Barthelemy and Haftka, 1993; Myers and Montgomery, 1995; Roux et al., 1998; Ruzika and Wiecek, 2003; Simpson et al., 2004). A typical approximation approach includes two components: (1) DOE, in which a sample of experiments (or observed design points) is selected according to a predefined criterion, and (2) metamodeling, in which the experiments are used to build a metamodel of the simulation (i.e., a compact or less costly model of the simulation model). Figure 1.1 represents an example in modeling, simulation and approximation for an engineering problem.



Engineering Problem

Computer Simulation

Approximation

Figure 1.1 The simulation and approximation for an engineering problem

According to how the metamodel interacts with a simulation and the optimizer, metamodeling approaches in the literature can be categorized as online and offline approaches. In an online metamodeling approach, metamodeling is made and adaptively improved in the course of optimization (Nair and Keane, 1998; Farina, 2001, 2002; Jin et al., 2001, 2002; Hong and Tahk, 2003; Nain and Deb, 2003). In offline metamodeling approaches, optimization is performed based on an a priori constructed metamodels (Papadrakakis et al., 1999; Wilson et al., 2001; Koch et al., 2002; Lian and Liou, 2004; Chung and Alonso, 2004; Fang et al., 2004). Figure 1.2 shows the schemes for online and offline metamodeling with an optimizer.

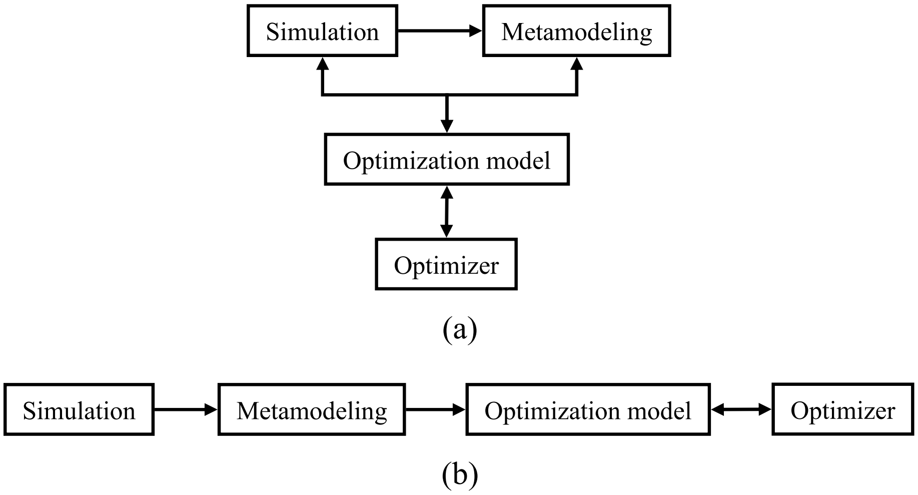


Figure 1.2 The schemes for (a) online metamodeling, and (b) offline metamodeling, with an optimizer

Offline metamodeling approaches often require fewer function calls than those for online metamodeling. However, offline metamodeling approaches can introduce false optimum solutions when used with multi-objective optimization and when it is difficult to obtain a good fidelity metamodel over the entire input space while at the same time use a low number of function calls (Simpson et al., 2001; Wilson et al., 2001). Online metamodeling approaches, on the other hand, can gradually improve the metamodel accuracy (Jin, 2005). Hence, both online metamodeling and offline approaches are useful in engineering design optimization. Online metamodeling approaches are often applied to the situation where relatively no limit exists for the number of function calls available, while offline metamodeling approaches are proper for the situation that a limited number of function calls are available. In this dissertation, both online and offline metamodeling techniques are explored in the context of genetic algorithm based design optimization methods.

The metamodel accuracy is always critical for both online and offline metamodeling approaches. In addition, the DOE technique used is very important for improving the metamodel accuracy as discussed in the next few paragraphs.

In a DOE technique, the experiments can be selected all-at-once or selected at different stages. The latter approach is often referred to as a sequential approach. Furthermore, a DOE approach is called sequentially adaptive if the information from both the last selected design points and the metamodel is used in selecting new design points. Figure 1.3(a) shows a scheme for non-adaptive DOE in which only the information from the selected design points is used for selecting new design points. Figure 1.3(b) shows a scheme for adaptive DOE, in which the information from both the selected design points and the corresponding metamodel is used for selecting new design points.

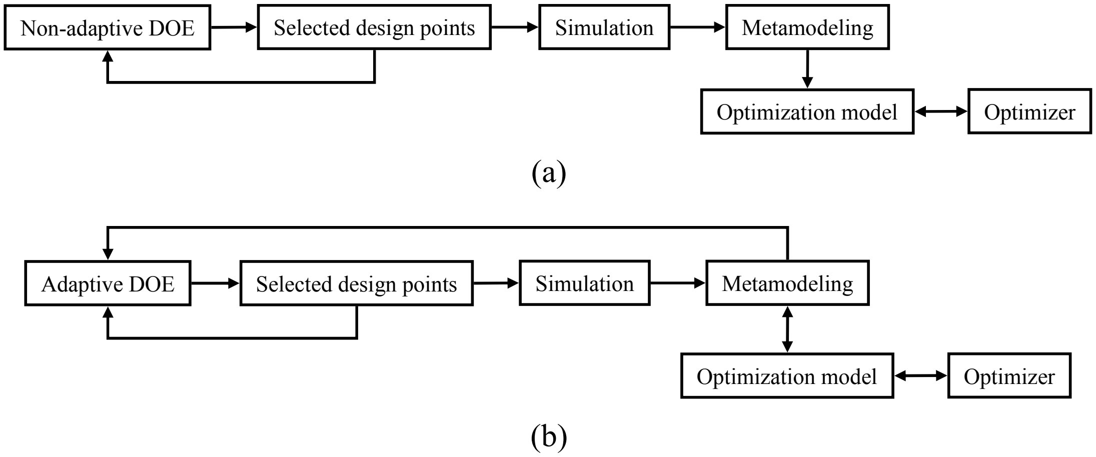


Figure 1.3 Optimization with (a) non-adaptive DOE, and (b) adaptive DOE

In general, adaptive DOE approaches are considered to be superior to non-adaptive approaches in terms of metamodel accuracy. Part of this dissertation focuses on the development and application of sequentially adaptive DOE methods and their integration with online and offline metamodeling approaches.

Although many DOE and metamodeling methods in the context of online metamodeling have been reported in the literature, most of them are subjectively devised. Moreover, the DOE and metamodeling approaches are often used separately in the literature. More function calls can be saved if they are used in an integrated manner. In the context of offline metamodeling, the research focus is to use as few as possible function calls and to build a reasonably accurate metamodel. However, most of the DOE methods do not use the available information efficiently. Finally, most of the current metamodeling approaches assume that the simulation model has a single response (output), while in reality a simulation model often produces multiple responses simultaneously.

The overall objective of this dissertation is to develop objective methods for DOE and metamodeling in the context of online and offline metamodeling. Also, a metamodeling technique for deterministic simulations with multiple responses will develop.

This overall objective aims at facilitating computer-aided engineering design optimization, particularly in the context of MOGAs, by reducing the number of function calls needed for convergence to an optimal solution.

1.2 Research Thrusts

To achieve the aforementioned overall objective, four research thrusts have been identified and pursued in this dissertation as follows: (1) Research Thrust 1: Online

Metamodeling for Fitness Estimation (see Chapter 3 for details), (2) Research Thrust 2: Adaptive DOE for Online Metamodeling (see Chapter 4 for details), (3) Research Thrust 3: Adaptive DOE for Offline Metamodel (see Chapter 5 for details), and (4) Research Thrust 4: Dependent Metamodeling for Multi-Response Simulations (see Chapter 6 for details).

A brief description of the motivation and objective for each research thrust is given in Sections 1.2.1 to 1.2.4.

1.2.1 Research Thrust 1: Online Metamodeling for Fitness Estimation

A number of techniques incorporating metamodeling with population-based optimization methods, such as GAs, have been reported in the literature (Jin, 2005). Some of these methods use metamodeling for fitness estimation and are referred to as online metamodeling (Chung and Alonso, 2004; Farina, 2001; Nain and Deb, 2002, 2003; Oduguwa and Roy, 2002). Note that the use of metamodeling for fitness evaluation can increase the risk of generating false optima (Jin, 2005), especially when the metamodel is not accurate in the early generations of a population-based optimization method. Moreover, the online metamodeling methods reported in the literature are all subjective: The points are chosen randomly or subjectively for metamodeling. Research Objective 1 addresses this issue.

Research Objective 1: *To develop a new online metamodeling approach that uses an objective criterion to determine whether a simulation model or its metamodel replacement should be used to evaluate design points.*

This new approach is referred to as online metamodeling for fitness estimation in the context of population based design optimization methods such as MOGAs.

A portion of this work was presented in Li et al. (2006b) and Li et al. (2007).

1.2.2 Research Thrust 2: Adaptive DOE for Online Metamodeling

Metamodeling can also be used for generating new points (Anderson and Hsu, 1999; Abboud and Schoenauer, 2001; Rasheed, 2000; Rasheed et al., 2005; Li et al., 2007). When metamodeling is used with population based optimization methods, the risk of generating false optima is avoided if metamodeling is used for generation or reproduction of new points (Shan and Wang, 2005). However, the use of metamodeling in reproduction can not reduce the number of function calls as many as it does in the fitness evaluation. Moreover, the savings obtained in terms of the number of function calls due to the use of metamodeling in fitness evaluation and reproduction of design points are independent since such savings are obtained during different stages in a population-based optimization method. Thus an integration of metamodeling with both fitness evaluation and reproduction should reduce the number of function calls even more compared to research thrust 1. Such an approach has not been reported in the literature. Research Objective 2 addresses this issue.

Research Objective 2: *To develop a new adaptive DOE method and integrate it with the online metamodeling of Research Thrust 1, so that the number of function calls can be further reduced. This new approach is referred to as adaptive DOE for online metamodeling.*

A portion of this research thrust was presented in Li et al. (2007).

1.2.3 Research Thrust 3: Adaptive DOE for Offline Metamodeling

In the context of offline metamodeling, DOE methods are critical for improving metamodel accuracy. Adaptive DOE approaches use the information from all existing experiments as well as the metamodel for selecting new designs. According to the adaptation mechanism, adaptive DOE approaches in the literature can be grouped into three types. These are based on: i) Being adaptive according to the estimated optima (maxima or minima) (e.g., Watson and Barnes, 1995), ii) being adaptive according to the location of design points in the input space with the largest estimated uncertainty associated with an estimated response (e.g., Cox and John, 1992, 1997), and iii) adaptation to the location of estimated optima with large uncertainty (e.g., Schonlau et al., 1997; Jones et al., 1998; Sasena, 2002; Sasena et al., 2000; Farhang-Mehr and Azarm, 2002, 2005; Farhang-Mehr et al., 2003). All the aforementioned approaches directly use estimated function responses and/or uncertainty in selecting additional points, while such estimates are often inaccurate especially in the early-stage of an optimization process. Hence, these adaptive DOE approaches may sample some points that have neither good objective function values nor large uncertainty associated with an estimated response. As a result, sampling in the previous methods consumes precious function calls without providing much insight or help in metamodel construction. Research Objective 3 addresses this issue.

<u>Research Objective 3</u>: *To develop a new adaptive DOE method for offline metamodeling that is objective and uses all the available information efficiently to improve the metamodel accuracy.*

This new approach is referred to as an accumulative error-based DOE approach. Also, a portion of this work was presented in Li and Azarm (2006).

1.2.4 Research Thrust 4: Dependent Metamodeling for Multi-Response Simulation

Engineering simulations can produce multiple responses. One can treat each of these responses as a separate 'model' and construct a metamodel for each response. This is a fairly common approach in the literature and is referred to as independent metamodeling. The previously reported independent metamodeling approaches are based on an independent treatment of responses and therefore do not exploit information such as correlations among responses. Hence, these independent metamodeling approaches can be inefficient.

The research on how to handle multiple responses in a computationally efficient manner and particularly in the context of more modern metamodeling techniques such as kriging remains sparse. Several researchers (e.g., Khuri, 1996; Chaio and Hamada, 2001) have partially addressed this issue by regression methods, using a "desirability" function to combine all responses from a simulation into one (e.g., Derringer and Suich, 1980; Del et al., 1996), or using generalized distance and dual responses to account for the correlation among responses (e.g., Vining and Myers, 1990; Lucas, 1994; Myers et al., 1997). Cokriging, an extension of kriging, has also been reported in the literature to handle multiple responses (e.g., Ver Hoef and Cressie, 1993; Romero et al., 2006). However, in cokriging it can be difficult to estimate cross-correlation parameters because the value of simulation responses and their gradients are required (Chung and Alonso, 2002). Furthermore, obtaining gradients may not be easy and can be time consuming. Research Objective 4 addresses the above mentioned issues.

Research Objective 4: *To develop a new metamodeling approach for multi-response simulations.*

The new approach is referred to as dependent metamodeling and uses all of the available information in building a metamodel for every response from a simulation simultaneously. Also, a portion of this work was presented in Li et al. (2006a).

1.3 Organization of Dissertation

The organization of the rest of the dissertation is as follows. In Chapter 2, the definition and terminology used in this dissertation are outline. In Chapter 3, a new method for online metamodeling for fitness estimation of population based optimization methods is presented. In Chapter 4, a novel adaptive DOE approach is developed and integrated with the online metamodeling approach presented in Chapter 3. After that, a new adaptive DOE approach for offline metamodeling is proposed in Chapter 5. Followed in Chapter 6, a novel dependent metamodeling approach is developed for multi-responses deterministic simulations. Finally, in Chapter 7, the dissertation is concluded with remarks and discussions for the research thrusts, as well as suggestions for future research directions. Figure 1.4 shows the organization and flow of information for the dissertation.

In the next chapter, the definition and terminology used throughout this dissertation are given.

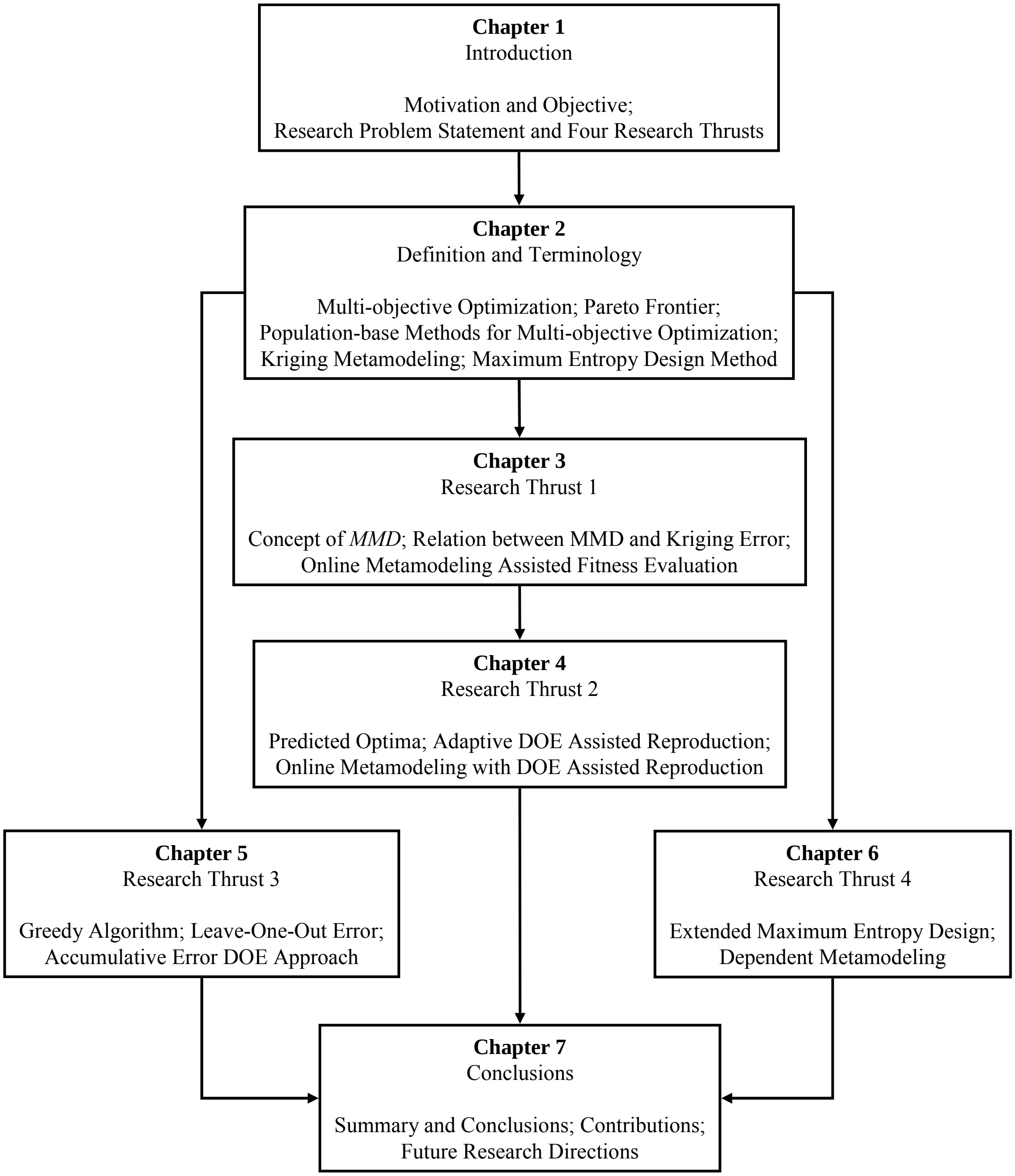

Chapter 1
Introduction
Motivation and Objective;
Research Problem Statement and Four Research Thrusts
Chapter 2
Definition and Terminology
Multi-objective Optimization; Pareto Frontier;
Population-base Methods for Multi-objective Optimization;
Kriging Metamodeling; Maximum Entropy Design Method
Chapter 3
Research Thrust 1
Concept of *MMD*; Relation between MMD and Kriging Error;
Online Metamodeling Assisted Fitness Evaluation
Chapter 4
Research Thrust 2
Predicted Optima; Adaptive DOE Assisted Reproduction;
Online Metamodeling with DOE Assisted Reproduction
Chapter 5
Research Thrust 3
Greedy Algorithm; Leave-One-Out Error;
Accumulative Error DOE Approach
Chapter 6
Research Thrust 4
Extended Maximum Entropy Design;
Dependent Metamodeling
Chapter 7
Conclusions
Summary and Conclusions; Contributions;
Future Research Directions

Figure 1.4 Organization of dissertation

Chapter 2 Definition and Terminology

2.1 Introduction

In this chapter, we provide several definitions and terminologies that will be used throughout the dissertation. First, the formulation for a general multiobjective optimization problem, based on which metamodeling is devised, will be introduced. Next, the concept of domination in the multi-objective optimization is introduced. Followed is a brief description of population-based optimization methods and the computational challenge when such methods are applied to multi-objective optimization problems. We then discuss kriging metamodeling that is used as the metamodeling technique in the research thrusts. After that, we introduce the basic concept behind the maximum entropy design method, which in later chapters is modified and used for DOE in the proposed approaches. Finally, we introduce the measures used for evaluating the accuracy of a metamodel.

2.2 Multi-Objective Optimization

A multi-objective optimization problem is formulated as follows:

$$\begin{array}{lll} \underset{\boldsymbol{x}}{\text{minimize}} & f_i(\boldsymbol{x}) & i = 1,\ldots,M \\ \text{subject to}: & g_j(\boldsymbol{x}) \leq 0 & j = 1,\ldots,J \\ & h_k(\boldsymbol{x}) = 0 & k = 1,...,K \end{array} \tag{2.1}$$

where f_i refers to an objective function, $\boldsymbol{x} = (x_1,\ldots,x_N)^T$ is a vector of design variables, the functions $g_1,...,g_J$ are the inequality constraints, and $h_1,...,h_K$ are the equality constraints. It is assumed that at least two of the M objective functions are conflicting with each other.

In the rest of the dissertation, $\boldsymbol{x}$ (or x if it has one dimension) is referred to as a *point*. The computer program representing all of f_i, g_j, and h_k together is defined as a *deterministic simulation*. A deterministic simulation is assumed to produce an identical response for the same input point. Also, a deterministic simulation may produce multiple responses in a single run. In this case, it is defined as a deterministic simulation with multiple responses. In the rest of the dissertation, a deterministic simulation is simply called a *simulation*. An evaluation of a point by a simulation is denoted as one *function call*. Note that the counting of function calls is based on the number of points evaluated. In other words, even if the simulation has multiple responses, the evaluation of a point for all the responses counts for one function call.

In the following, some further definitions are given in the context of a simulation.

2.2.1 Input space

A space Ω formed by all possible points to a simulation is referred to as the *input space* for the simulation. A point in the input space can be represented by a vector. Each element of this vector can be continuous or discrete. Each element is assumed to be bounded by lower and upper bounds and thus can be normalized in the range: [0, 1].

2.2.2 Experiment

A point is referred to as an *experiment* if it is evaluated for its true response by a simulation. In contrast, a point that is not evaluated by the simulation is called an unobserved point and denoted as x_0.

2.2.3 Design

A set of experiments is referred to as a *design* $\boldsymbol{D}$. The true responses for all experiments in a design are obtained by running the simulation.

2.3 Pareto Frontier

The optimum solutions for a multi-objective optimization problem, Eq. (2.1), form a Pareto frontier in the objective space (i.e., *f*-space). Solutions along a Pareto frontier are non-dominated. A point is said to be non-dominated if no other point in the current set of points is better than that point with respect to all objectives (Goldberg, 1989; Deb, 2001). In this way, the domination status of a point can be determined as either being dominated or non-dominated. The set of all non-dominated points in the current set of points forms a non-dominated design set. The remaining points form a dominated design set. In this regard, the Pareto frontier is a set of points that are non-dominated over the entire design space.

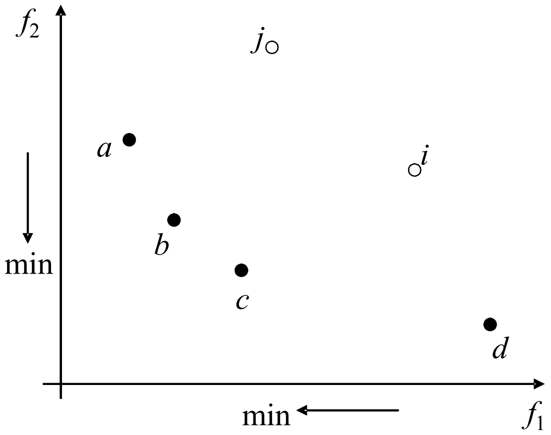


Figure 2.1 Pareto dominance in multi-objective optimization

Figure 2.1 above shows an example of dominance in objective space, whereby both objectives are being minimized (as illustrated by the two arrows on the two axes), with a population of six points. Four of these points {*a*, *b*, *c*, *d*} are non-dominated. That is, each of these points is not dominated in terms of all objective function values by any other point in the population. So, in this population, points {*a*, *b*, *c*, *d*} form the non-dominated set, and the points $\{i, j\}$ form the dominated set.

It can be observed in Figure 2.1 that the domination status of points is dependent on their relative position in the objective space. However, the domination status is not very sensitive to the change of the position of a point. For example in Figure 2.2 below, even if the point *j* changes its position to *j'*, it is still dominated (i.e., by points *a*, *b* and *c*). This observation indicates that a point can have some degree of variance in terms of its fitness (i.e., position in *f*-space) without changing the domination status with respect to other points in the population.

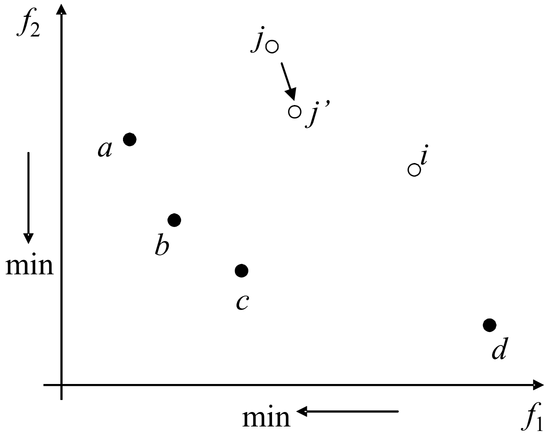


Figure 2.2 Domination status when the position of an point changes

2.4 Population-Based Methods for Multi-Objective Optimization

Generally speaking, there are two classes of design optimization methods that can be used to solve optimization problems (e.g., Belegundu and Chandrupatla, 1999; Papalambros and Wilde, 2000; Arora, 2004). These two classes are gradient- and non-gradient-based methods (e.g., population-based optimization methods). Gradient-based methods (e.g., Bazaraa et al., 1993) require derivative information for the optimization functions (i.e., objective and constraint functions) and usually have an implicit assumption that these functions are "smooth" and that design variables are continuous.

In general, gradient-based methods can only guarantee a local optimum design solution unless the functions used in the problem have special properties such as linearity or convexity.

The smoothness assumption can be relaxed in population-based optimization methods such as evolutionary algorithms (e.g., Rosenberg, 1967; Holland, 1975; Goldberg, 1989; Deb, 2001) and particle swarm optimizers (e.g., Kennedy and Eberhart, 1995; Li, 2003; Coello et al., 2004). As a result, population-based optimization methods have become increasingly popular for solving multiobjective optimization problems due to the generic ability of these methods to handle complex real-world problems. Since such methods maintain a population of design points, this allows exploration of different parts of the solution space simultaneously.

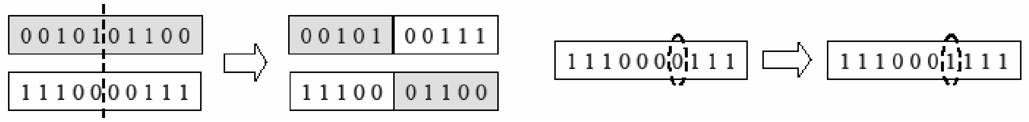
Various versions of the aforementioned population-based optimization methods have been reported (e.g., Schaffer, 1984; Deb, 1999; Coello, 2000; Fonseca and Fleming, 1993; Srinivas and Deb, 1994; Deb et al., 2000; Norn and Nafpliotis, 1993; Zitzler and Thiele, 1998; Li, 2003; Coello et al., 2004; Alvarez-Benitez et al., 2005; Sierra and Coello, 2005; Reddy and Kumar, 2007). These methods have the following common characteristic: They all use i) some sort of algorithm based on the concept of dominance to determine the fitness for a point, ii) operators for generating points so that local optima can be avoided, and iii) numerous fitness evaluations (i.e., simulation or function calls) to obtain optimum solutions. This large number of function calls may not be affordable in practice despite recent and foreseeable dramatic improvements in computer hardware and solution algorithms. One of the strategies to reduce the computational cost is to use metamodeling to partially substitute the simulation with a metamodel during the optimization process.

In this dissertation, a MOGA is used as the population based optimization method. In particular, the approaches of Chapter 3 and Chapter 4 are implemented with MOGA. Note that the proposed approaches are not restricted to this specific optimization method, although they implemented using this method. Actually, the proposed approach can be used with any population-based optimization method as long as the concept of dominance is used for fitness evaluation.

In the next subsection, a brief introduction of MOGA is given.

2.4.1 Multi-Objective Genetic Algorithm

MOGA is based on the GA that is extended for multi-objective optimization. GA was developed for global optimization of unconstrained single-objective optimization problems. GAs are stochastic and population based optimization methods whereby the input space Ω is searched through selected points using genetic operations. In a GA, each design variable is coded as a binary string and then all design variables are collected to form a chromosome representing a point. Each point is assigned with a fitness value based on the objective function value to present the goodness of this point. GA creates a new generation of points through two important operations, i.e., crossover and mutation. In a crossover operation, as shown in Figure 2.3(a), two points are randomly selected as parents and then portions of their chromosomes are swapped to produce offspring points. For diversity, as shown in Figure 2.3(b), a mutation operation is performed in that a bit is randomly selected and changed from 0 to 1 or vice versa.



(a) (b)

Figure 2.3 (a) Crossover, and (b) mutation operation

In GAs, points are also “inherited” in that a subset of points which have good fitness values are chosen to migrate into the next generation. Together crossover, mutation, and inheritance provide a powerful search scheme for finding a global optimum point (Deb, 2001).

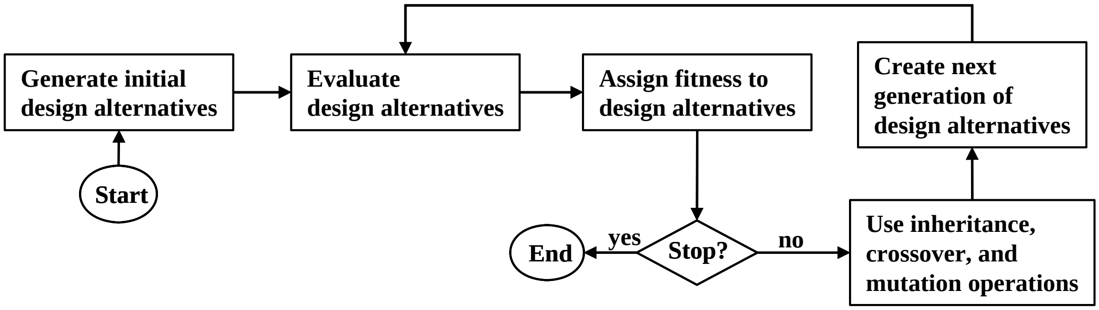

Generate initial design alternatives
Evaluate design alternatives
Assign fitness to design alternatives
Create next generation of design alternatives
Start
End
yes
Stop?
no
Use inheritance, crossover, and mutation operations

Figure 2.4 Steps for a GA

As shown in Figure 2.4, a GA starts with an initial population of points that are randomly generated. Next, points in the population are evaluated and their fitness assigned. After that, a stopping criterion is checked to see whether the fitness of points can be further improved or whether a predefined number of iterations have been reached. If the stopping criterion is satisfied, then the GA is stopped; otherwise, crossover, mutation, and inheritance operations are applied to produce a new generation of points and the procedure is repeated.

GAs can be easily modified to handle multi-objective optimization problems. The multi-objective optimization version of the GA is referred to as a MOGA (e.g., Narayanan and Azarm, 1999; Deb 2001; Kurpati et al., 2002). MOGAs are capable of handling multiple objectives by defining fitness in a Pareto optimality sense as shown in Figure 2.1. For instance, Deb (2001) proposed a MOGA with a non-dominated sorting scheme, in which the domination status among points are established pair-wise for each point. Conventional MOGAs can also handle multi-objective optimization problems with constraints. A common approach for handling constraints is the penalty method (e.g., Coit et al., 1996; Kurpati et al., 2002) whereby the constraints are combined with the objective functions as a penalty when they are not satified at a point.

In this dissertation, a MOGA that uses non-dominated sorting scheme (Deb, 2001) with an elitism strategy (Li et al., 2006b) for fitness evaluation and a penalty method (Kurpati et al., 2002) for constraint-handling is referred to as a *conventional MOGA*. A flowchart for a MOGA is given in Figure 2.5.

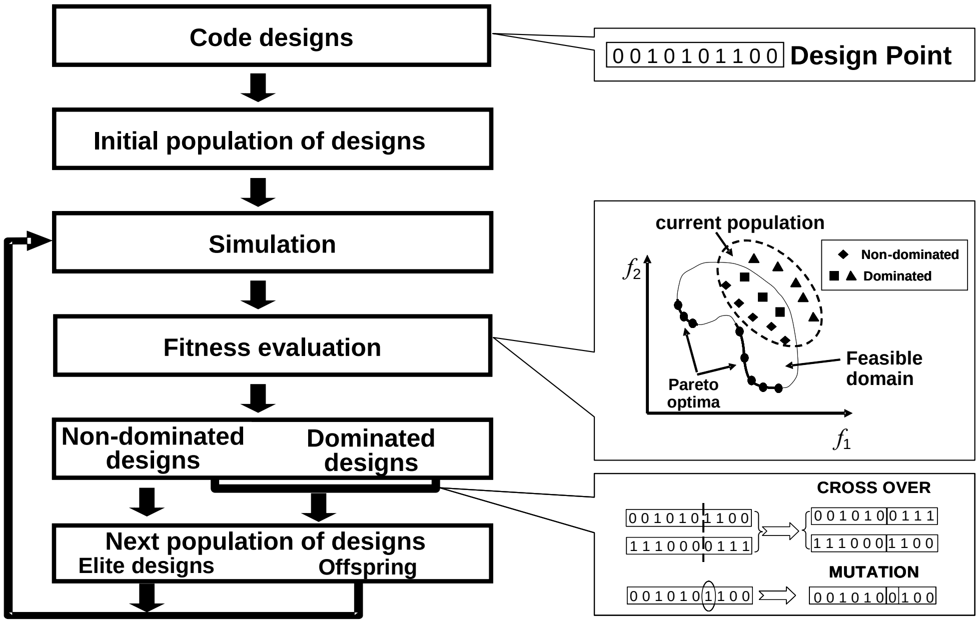


Figure 2.5 Flowchart for a MOGA

It can be seen that the major steps in a MOGA as shown in Figure 2.5 are similar to those in a GA as shown in Figure 2.4. Note that the fitness evaluation in a MOGA is performed by obtaining non-dominated and dominated sets in the current population as shown in the middle block on the right part in Figure 2.5. The other two blocks on the right part Figure 2.5 show the coding of a point and the crossover and mutation operations in a MOGA, separately.

MOGAs are considered robust and capable of finding global optima. However, as a population based optimization method, a MOGA requires a large number of function calls that has prohibited applications of these algorithms to realistic engineering design problems.

As discussed, a common way to reduce the number of function calls is to use a metamodel to partially substitute the intensive simulation in the optimization process. In the next section, kriging metamodeling is introduced, which is used in the rest of the dissertation as the metamodeling technique.

2.5 Kriging Metamodeling

Kriging is an interpolative metamodeling method for response approximation from a simulation (e.g., Sacks et al., 1989; Jones, 2001). Over the past several years kriging has become popular for metamodeling in the field of engineering design because of its general predictive modeling ability.

Kriging can be viewed as a linear predictor estimating unknown response values based on all observed response values. To simplify the description of kriging, we consider a single-input (x) single-response (y) simulation model, that is, $y = f(x)$. A kriging method treats the deterministic response y of the simulation as a realization of a stochastic process Y:

$$Y = \mu + Z(x) \tag{2.2}$$

where Y is the approximation of a true response y, μ is a constant representing the known prior information (i.e., mean of the responses based on the current metamodel), x a point, and Z is assumed to be a stochastic process with zero mean and non-zero variance σ^2. It is assumed that the function f to be approximated is continuous and thus the responses will not change significantly for two adjacent input points x. In practice, this assumption can be somewhat relaxed and replaced with a weaker assumption, i.e., for two points x_i and x_j, the response values of $y(x_i)$ and $y(x_j)$ should be close to each other if the distance between x_i and x_j is small. In other words, one can state that the random variables $Y(x_i)$ and $Y(x_j)$ tend to be highly correlated if the distance between x_i and x_j is small. The correlation "*Corr*" between any two random variables $Y(x_i)$ and $Y(x_j)$ can be calculated by a variety of correlation functions (e.g., Clark et al., 2005). The correlation function presented by Sacks et al. (1989) is the most commonly used:

$$Corr[Y(x_i), Y(x_j)] = \exp(-\theta \| x_i - x_j \|) \tag{2.3}$$

where θ is a parameter determined by the degree of the correlation among the responses. The larger the value of θ, the weaker the correlation, and vice versa.

Suppose there are n experiments in the current design $\boldsymbol{D}$: $x_1, x_2, \ldots, x_n$, and $\boldsymbol{y}$ represents the set of n corresponding observed actual responses:

$$\boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \tag{2.4}$$

The covariance matrix **Cov** of $\boldsymbol{y}$ can be defined by

$$\mathbf{Cov}(\boldsymbol{y}) = \sigma^2 \boldsymbol{R} \tag{2.5}$$

where $\boldsymbol{R}$ is a $n \times n$ correlation matrix with the (i, j) element given by Eq. (3.3):

$$\boldsymbol{R} = \begin{bmatrix} Corr(Y(x_1),Y(x_1)),...,Corr(Y(x_1),Y(x_n)) \\ \cdots \\ Corr(Y(x_N),Y(x_1)),...,Corr(Y(x_n),Y(x_n)) \end{bmatrix} \tag{2.6}$$

The diagonal elements of ***R*** is always of the form Corr($Y(x_i)$,$Y(x_i)$) and are equal to one. Also, define ***r*** as a $n \times 1$ correlation matrix for ***y*** and the estimated response for an unobserved point x_0, and ***J*** is a $n \times 1$ vector of ones:

$$\boldsymbol{r} = \begin{bmatrix} Corr(Y(x_0),Y(x_1)) \\ \vdots \\ Corr(Y(x_0),Y(x_n)) \end{bmatrix} \tag{2.7}$$

So far we have assumed that μ and σ^2 are known a priori. In fact, they can also be estimated after *n* experiments as follows:

$$\hat{\mu} = \frac{\boldsymbol{J}^T \boldsymbol{R}^{-1} \boldsymbol{y}}{\boldsymbol{J}^T \boldsymbol{R}^{-1} \boldsymbol{J}} \tag{2.8}$$

$$\hat{\sigma}^2 = \frac{(\boldsymbol{y} - \boldsymbol{J}\hat{\mu})^T \boldsymbol{R}^{-1} (\boldsymbol{y} - \boldsymbol{J}\hat{\mu})}{n} \tag{2.9}$$

where $\hat{\mu}$ and $\hat{\sigma}^2$ are estimates for μ and σ^2, respectively. Then the kriging estimates the response and the corresponding variance for an unobserved point x_0 as follows:

$$Y(x_0) = \hat{\mu} + \boldsymbol{r}^T \boldsymbol{R}^{-1} (\boldsymbol{y} - \boldsymbol{J}\hat{\mu}) \tag{2.10}$$

$$s^2(x_0) = \hat{\sigma}^2 \left[1 - \boldsymbol{r}^T \boldsymbol{R}^{-1} \boldsymbol{r} + \frac{(1 - \boldsymbol{r}^T \boldsymbol{R}^{-1} \boldsymbol{r})^2}{\boldsymbol{J}^T \boldsymbol{R}^{-1} \boldsymbol{J}} \right] \tag{2.11}$$

where $Y(x_0)$ is the estimated response and $s^2(x_0)$ is the corresponding kriging variance, which serves as a measure of uncertainty for an estimated response. Clearly, for observed experiments, the kriging variance s^2 is zero. On the other hand, there is a non-zero kriging variance for any unobserved point (i.e., x_0). The further away is x_0

from current existing experiments, the higher the value of the kriging variance.

When Eq. (2.2) is assumed to be a normal process, the kriging variance is a conditional normal distribution with zero mean and $s^2(x_0)$ as the variance [US Army Corps, 1997]. The kriging variance can be converted to a "prediction interval" with a 95% confidence level and the range of the interval equals to $4s(x_0)$ (or $\pm 2s(x_0)$), as shown in Figure 2.6.

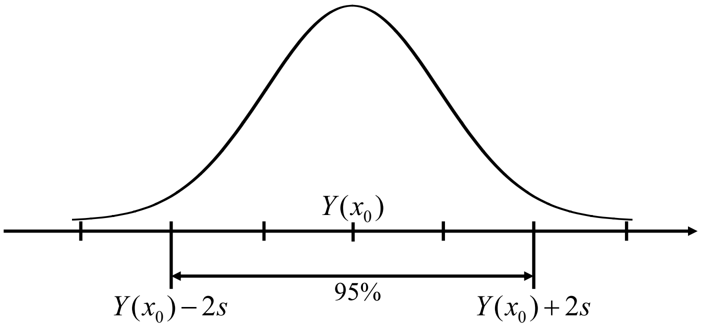


Figure 2.6 Prediction interval for the kriging variance

We define the prediction interval for an unobserved point x_0 as $I_P(x_0)$ for a 95% confidence level:

$$I_P(x_0) = 4s(x_0) \tag{2.12}$$

This prediction interval can be used to measure the uncertainty of a response obtained by a kriging metamodel.

The kriging metamodeling is also illustrated in Figure 2.7. Figure 2.7(a) shows the prior distributions of *Y*'s with the given mean of the stochastic process and without any observation of the response. On the other hand, Figure 2.7(b) shows the posterior distributions based on observing the responses for two experiments (x_1 and x_i). The hollow points in Figure 2.7(a) indicate that these points are not observed. The estimated responses are represented by *Y* (e.g., Y_2 for x_2 and Y_0 for x_0 as shown in

Figure 2.7(a)). The two points x_1 and x_i are observed (shown as solid points as represented with y_1 and y_i in Figure 2.7(b)) and the kriging method updates the estimates of other responses, i.e., it revises the prior distribution of Y's. It can be observed that the estimated responses in Figure 2.7(a) and Figure 2.7(b) for Y_2 are different. The change of the kriging variance is also illustrated in Figure 2.7.

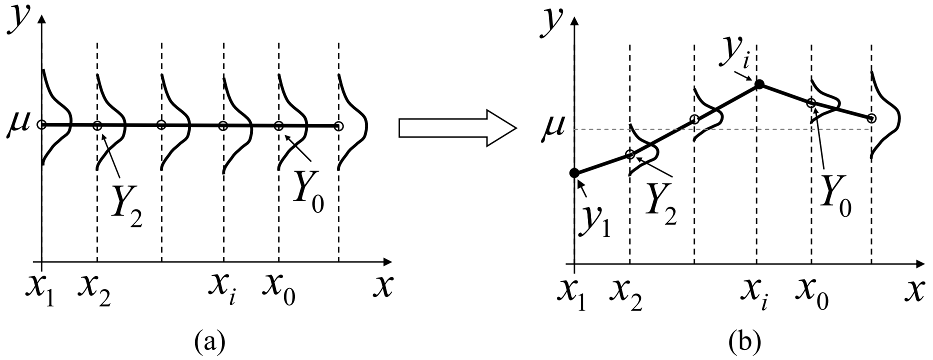


Figure 2.7 Kriging metamodeling: (a) before any experiment is observed, and (b) after two experiments (x_1 and x_i) are observed

2.6 Maximum Entropy Design Method

Lindley (1956) introduced Shannon's entropy H (Shannon, 1948) into the field of DOE and interpreted it as the amount of information retrieved from an experiment. Currin et al. (1988, 1991) then proved that the entropy criterion H selects the new experiment x_{i+1} (assuming a scalar input) as the one that maximizes the expected retrieved information due to the new experiment, i.e.,

$$x_{i+1} = \arg\max H(x_1, x_2, \ldots, x_i\,; x) \tag{2.13}$$

where $(x_1, x_2, \ldots, x_i)$ are for the existing i experiments, and "arg max" denotes the optimal solution, e.g., x_{i+1}, of the maximum entropy H problem. x_{i+1} is for the new experiment to be found by the method.

Later, it was further proved (e.g., Shewry and Wynn 1987) that assuming normal priors, this criterion is equivalent to maximizing the determinant "*det*" of the covariance matrix ***R*** as follows:

$$x_{i+1} = \text{arg max } det(\boldsymbol{R}) \tag{2.14}$$

where ***R*** is (defined in Eq. 2.6) a $(n+1)\times(n+1)$ matrix with each element calculated using $(x_1, x_2, \ldots, x_i, x_{i+1})$.

Using the maximum entropy design, the resulting design is almost symmetric with emphasize on the boundaries. For instance, Figure 2.8 shows a design ***D*** obtained using the maximum entropy design with 13 experiments in a two dimensional input space.

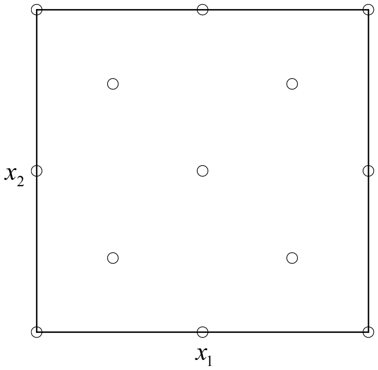


Figure 2.8 Maximum entropy design for 13 experiments

Clearly, this approach uses no information from the response for sampling subsequent experiments, which makes it appropriate for initial design selection where no prior information is available. Moreover, the approach emphasizes selecting points in the input space where fewer experiments have been sampled.

2.7 Measurement for Metamodel Accuracy

In order to assess a metamodel accuracy, a randomly selected set of test points is used and the Root Mean Squared Error (RMSE) and the Maximum Absolute Error (MAE) can be calculated for this set.

The RMSE, which calculates the overall accuracy of a metamodel in the response (or output) space, is obtained by:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left[\hat{y}(x_i) - y(x_i)\right]^{2}} \tag{2.17}$$

where x_i is an experiment in the testing sample, $\hat{y}(x_i)$ is an estimated response for x_i, $y(x_i)$ is the actual response, and n is the number of the test points used.

The MAE, which calculates the largest difference between a metamodel and corresponding actual response for all the test points, is obtained by:

$$\text{MAE} = \max\left|\hat{y}(x_i) - y(x_i)\right| \tag{2.18}$$

where "max" denotes the maximum difference for the estimated response and actual response obtained for all the test points.

In Chapter 5 and Chapter 6 of the dissertation, the quantities RMSE and/or MAE are used for the measurement of the accuracy for the obtained metamodels. Normalized values for the responses are used in calculating RMSE and MAE. Assuming that the lower bound and upper bound of a response is known (or known approximately), a normalized value (in a range of [0, 1]) for a response value (or its estimate) is obtained as follows:

$$\text{normalized value} = \frac{\text{response value - lower bound}}{\text{upper bound - lower bound}} \tag{2.19}$$

Note that the quantities RMSE and MAE are obtained for each response metamodel separately, thus the multiple responses with different scales will not influence metamodel accuracy assessment.

In the next chapter, the research thrust one, a new online metamodeling assisted optimization approach, is presented.

Chapter 3 Online Metamodeling for Fitness Evaluation

3.1 Introduction

A significant challenge in the applications of population based optimization methods to engineering design problems has been the high computational cost of these methods due to the large number of function calls required for these methods (Deb, 2001). As discussed before, a common strategy to reduce the computational effort for such optimization methods is to use metamodeling techniques. Researchers have been quite active in developing models and methods that improve the efficiency of population based optimization methods in terms of the number of function calls (Papadrakakis et al., 1999; Farina, 2001; Simpson et al., 2001; Jin et al., 2002; Hong et al., 2003; Jin, 2005; Wang and Shan, 2007).

The online metamodeling techniques are frequently used to reduce the number of function calls (Farina, 2001; Hong et al., 2003; Jin et al., 2002). Most of the online metamodeling approaches reported in the literature so far are focused on single-objective optimization. The research on how to embed metamodeling within multi-objective population based optimization methods remains sparse (Farina, 2001, 2002; Nain et al., 2003). Moreover, most of this type of approaches utilize neural network, which is well known to require a large number of function (simulation) calls (Simpson et al., 2001). Another unresolved issue in the current online metamodeling approaches is how to objectively decide when to switch to the metamodel during the optimization (Jin et al., 2001, 2002; Jin, 2005). Usually the switching between the original simulation model and the metamodel has been subjectively decided (Nain et al., 2003).

Finally, the fidelity of the metamodel may vary significantly during the optimization process and this can cause oscillation (Jin, 2005).

In this chapter we develop a novel online metamodeling approach to address all of the above mentioned shortcomings. More specifically, a new online kriging metamodeling assisted fitness evaluation approach is presented. We use an objective criterion to determine whether a simulation or its kriging metamodel replacement should be used to evaluate points. In the proposed approach, the kriging metamodels for the simulation are built and adaptively improved within a population based optimization method (i.e., MOGA). The approach is general and requires no additional function calls prior to the start of the optimization procedure to build the kriging metamodels offline. The proposed criterion is developed based on the metamodel's predicted error which can be easily obtained as a byproduct from kriging and without extra computation.

The rest of the chapter is organized as follows. In Section 3.2, the general framework for the online metamodeling approach is briefly introduced. In Section 3.3 the details of the proposed online metamodeling approach, namely kriging metamodeling assisted fitness evaluation, are given. Examples and corresponding results are discussed in Section 3.4. Finally, in Section 3.5 the main observations for the proposed metamodeling based MOGA approach are summarized.

A portion of this chapter was presented in Li et al. (2006b) and Li et al. (2007).

3.2 Online Metamodeling Framework

In online metamodeling based approaches, in the initial stages of a population based optimization method, rough metamodels are constructed. These metamodels are then gradually improved as more simulation data becomes available.

In Figure 3.1, a basic scheme for an online metamodeling approach during a single generation, *i*th population, of a population based optimization method is given. In this chapter, MOGA (See, e.g., Section 2.4.1) is used as the population based optimization method.

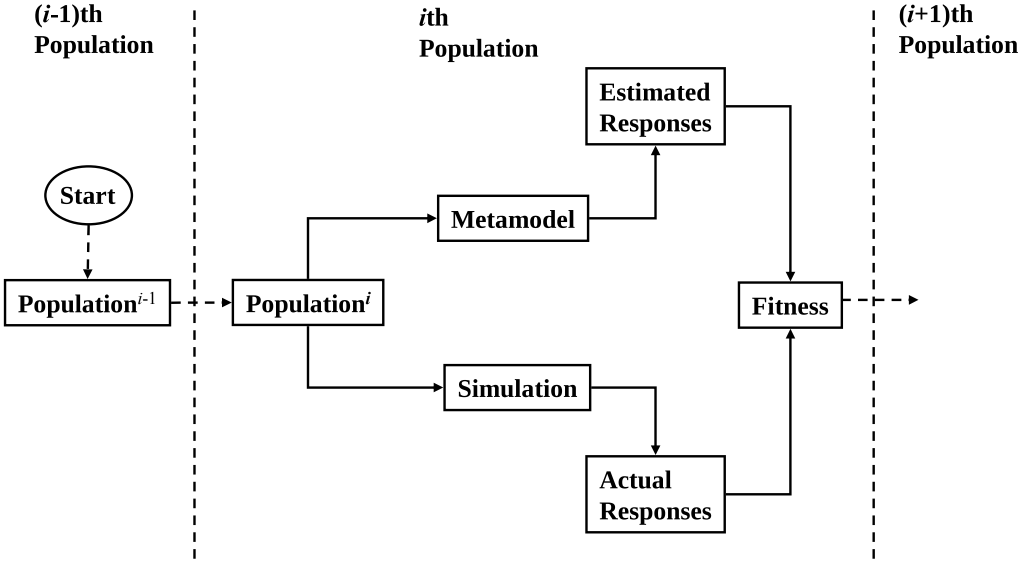


Figure 3.1 Online metamodeling assisted fitness evaluation in one generation

It can be seen in Figure 3.1 that the key issue is to determine a condition for using metamodels, during the optimization procedure, to partially substitute simulations. In other words, metamodeling is used to estimate the fitness for some points in the *i*th population, and simulations are used for other points.

3.3 Kriging Metamodeling Assisted Fitness Evaluation

The basic idea behind the proposed approach for metamodeling assisted fitness evaluation is to ensure (as much as possible) that in each generation the non-dominated set obtained using metamodels remains the same as the one obtained using the simulation. As discussed in Section 2.3, fitness values can have some variances without changing the dominance status. Thus, the challenge of using a metamodel or simulation can be tackled by determining how big the variance for a fitness value can be so that the dominance status will not be changed. Also, in kriging metamodeling a prediction interval (i.e., as denoted in Eq. (2.12)) can be obtained for each estimated response. That is, the variance of a fitness value can be obtained when kriging metamodeling is used. In this regard, the kriging is used as the metamodeling technique for the proposed approach.

3.3.1 Minimum of Minimum Distances (*MMD*)

In any generation, except in the initial population where all points are evaluated for their fitness values using the simulation, the kriging metamodel can be used to estimate the fitness values of the points. Based on these predicted values, the dominance status can be determined as follows in Figure 3.2.

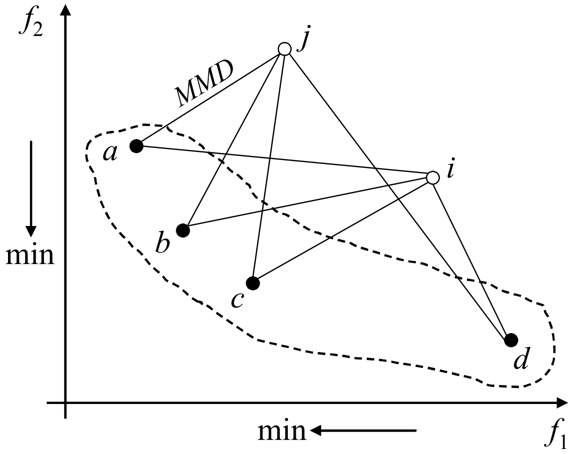


Figure 3.2 *MMD* in objective space

As shown in Figure 3.2, points $\{a, b, c, d\}$ form the estimated non-dominated set, while points $\{i, j\}$ form the dominated set. Then, the current population is divided into two sets, i.e., estimated dominated set and estimated non-dominated set. Note that this partitioning is based on the fitness values obtained from the kriging metamodel and no simulation calls (actual calls) are used at this stage. Furthermore, the quantity *MMD* is defined in the objective space as the minimum distance between all pairs of non-dominated points and dominated points as described next.

After the points in the current population are split into two sets: Estimated non-dominated and estimated dominated points, *MMD* in Figure 3.2 can be calculated using the follows equation:

$$\begin{aligned} MMD &= \min\{\|f(p)-f(q)\|_2\} \\ & p = a,b,c,d \ \text{(non-dominated points)} \\ & q = i,j \ \text{(dominated points)} \end{aligned} \tag{3.1}$$

where the norm is defined in the *f*-space (f_1, f_2, …, f_M). In order to connect *MMD* with prediction intervals of the kriging metamodel for each objective (recall Section 2.5), *MMD* is projected along each objective function axis to obtain MMD_{f_m}, m = 1, ..., M, as shown in Figure 3.3:

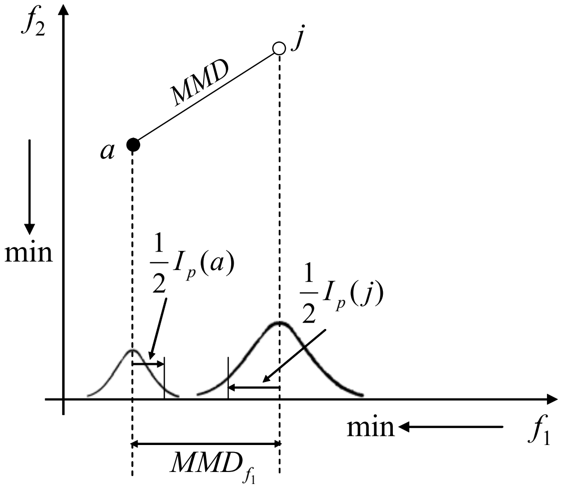


Figure 3.3 Projection of *MMD*

where MMD_{f_1} is the *MMD* projection along f_1, and $I_P(a)$ and $I_P(j)$ are prediction intervals for points *a* and *j*, separately.

3.3.2 Relation between *MMD* and Prediction Interval

As mentioned before, the estimated response from a kriging metamodel has a variance that can be converted to a prediction interval. As long as the values of a prediction interval do not affect the domination status of two points, the kriging metamodel outputs are considered accurate estimates for the true responses for these two points. Hence, the criterion that relates *MMD* and prediction interval can be defined by examining whether or not the domination status is changed when kriging estimates are used. If the domination status is changed, then the points that contribute to this change are evaluated using the original simulations. Those points that do not contribute to the change of the domination status can be estimated using kriging metamodels.

To obtain the relation between *MMD* and the prediction interval, for simplicity, we examine the situation in one dimension in the objective space (e.g., along objective function f_1 as shown in Figure 3.3). Because the prediction interval is symmetrical, we only need to examine one half of the interval for each point involved in the evaluation. It can be observed from Figure 3.3 that as long as the sum of half prediction intervals for points a and j do not exceed the corresponding *MMD* projection, the domination status of a and j along objective f_1 will not be changed even if the kriging estimates for both points are used. This relation between *MMD* and the prediction interval can also be expressed by the following condition:

$$I_p(a) + I_p(j) \leq 2MMD_{f_1} \tag{3.2}$$

Thus, Eq. (3.2) is used as the condition to determine whether kriging metamodels or simulations should be used for fitness evaluation. It can be seen from Eq. (3.2) that if the value of the half of the prediction interval of a point is greater than *MMD*, this point must be evaluated using the simulation. Note that this relation should hold for any point that can be estimated using kriging metamodels so that the domination status of all points in the current population is unchanged. For those points that can be estimated using kriging metamodels, the kriging-based estimated responses are considered to be accurate enough to ensure a correct domination status in the current population. However, it is going to require a lot of computation efforts if all the points in the current population are compared pair wise according to Eq. (3.2). Thus, an algorithm that can reduce the computation effort is proposed in the next section.

3.3.3 Shrinking, Sorting, and Selecting Algorithm

The aforementioned condition, Eq. (3.2), can be implemented in a three-stage procedure: Shrinking, sorting, and selecting stages. This procedure is applied in each generation of the proposed approach to determine the number of points that can be estimated using kriging metamodels. A description of the three-stage procedure follows. To simplify the exposition, we consider a single-input or single variable (x) case. The procedure can be readily extended to a multi-input case, as demonstrated by the examples later.

In the first stage (i.e., the shrinking stage), points with zero prediction interval I_p and points with a large value of the prediction interval (i.e., points with prediction interval greater than twice the value of *MMD*, recall Eq. (3.2)) are removed from the population. Note that the points with zero prediction have already been evaluated while those with a large value of the prediction interval are to be evaluated by the simulation. After the removal of these two kinds of points, the size of the population becomes smaller than the original population. We refer to this reduced size population as a “shrunk population”. We define the population size as n and the shrunk population size as n_s. Since some points in the current population are inherited, the number of points with a zero prediction interval is always greater than zero. That is, $n_s < n$, and we will work on a set with fewer points than the original population.

In the second stage (i.e., sorting stage), the points in the shrunk population are sorted based on their prediction interval values, from large to small. In other words, the first point x_1 in the set has the largest value of the prediction interval $I_p(x_1)$ while the last point x_{n_s} has the smallest value of the prediction interval $I_p(x_{n_s})$, as shown in the following in Eq. (3.3).

$$2MMD_{f_1} \geq I_p(x_1) \geq \ldots \geq I_p(x_{n_s}) > 0 \tag{3.3}$$

where x_i is an point in the sorted shrunk population, $i = 1,\ldots,n_s$.

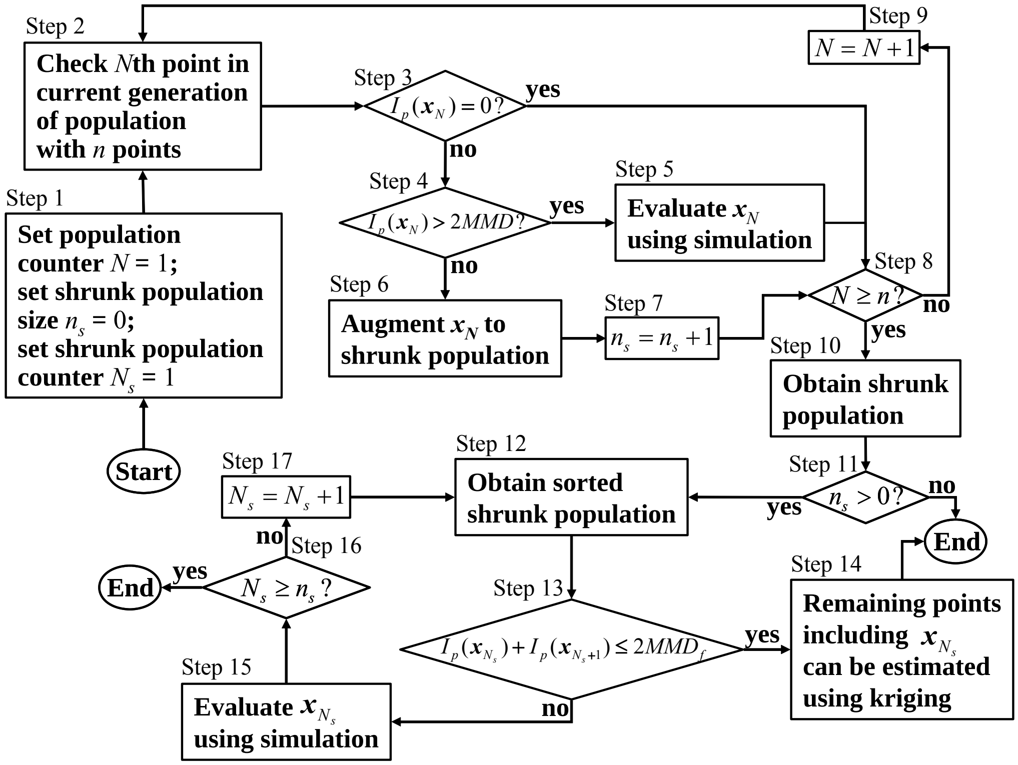
In the last stage (i.e., selecting stage), the points in the sorted shrunk population are examined, beginning from the first point x_1, to determine whether or not there exists a point x_i that satisfies the following condition, which is derived from Eq. (3.2):

$$I_p(x_i) + I_p(x_{i+1}) \leq 2MMD_{f_1} \tag{3.4}$$

If the condition in Eq. (3.4) is not satisfied for a point x_i, then x_i is removed from the sorted shrunk population and evaluated using the simulation. On the other hand, if the condition in Eq. (3.4) is satisfied, then following Eq. (3.3) we have:

$$2MMD \geq I_p(x_i) + I_p(x_{i+1}) \geq I_p(x_i) + I_p(x_{i+2}) \geq \ldots \geq I_p(x_i) + I_p(x_{i+n_s-1}) \tag{3.5}$$

which implies that the point x_i as well as all remaining points in the sorted shrunk population meet the condition in Eq. (3.2) and they can be estimated using the kriging metamodel for their fitness values. This three-stage procedure is presented in detail in a flowchart in Figure 3.4, where *N* is the counter for the original population (*N* = 1, …, *n*), and N_s is the counter for the shrunk population (N_s = 1, …, n_s).



Step 1
Set population counter N = 1; set shrunk population size n_s = 0; set shrunk population counter N_s = 1
Start
Step 2
Check Nth point in current generation of population with n points
Step 3
$I_p(\boldsymbol{x}_N) = 0?$
yes
no
Step 4
$I_p(\boldsymbol{x}_N) > 2MMD?$
yes
no
Step 5
Evaluate x_N using simulation
Step 6
Augment x_N to shrunk population
Step 7
$n_s = n_s + 1$
Step 8
$N \geq n?$
no
yes
Step 9
$N = N + 1$
Step 10
Obtain shrunk population
Step 11
$n_s > 0?$
yes
no
End
Step 12
Obtain sorted shrunk population
Step 13
$I_p(\boldsymbol{x}_{N_s}) + I_p(\boldsymbol{x}_{N_s+1}) \leq 2MMD_f$
yes
no
Step 14
Remaining points including $\boldsymbol{x}_{N_s}$ can be estimated using kriging
Step 15
Evaluate $\boldsymbol{x}_{N_s}$ using simulation
Step 16
$N_s \geq n_s?$
yes
no
End
Step 17
$N_s = N_s + 1$

Figure 3.4 Shrinking, sorting, and selecting algorithm

A step-by-step description of the algorithm is given as follows.

Step 1: Initialize counters: $N = 1$, $n_s = 0$, $N_s = 1$.

Step 2: Examine the value of prediction interval $I_p(x_N)$ of the *N*th point in current generation.

Step 3: If the value of prediction interval $I_p(x_N)$ for the *N*th point is zero (i.e., this point has been evaluated using simulation), go to Step 8.

Step 4: If the value of prediction interval $I_p(x_N)$ for the *N*th point is greater than

2*MMD* (i.e., this point has a large variance and needs to be evaluated using simulation), go to Step 5.

Step 5: Evaluate this point using simulation, and go to Step 8.

Step 6: If conditions in Step 3 and 4 are not met, augment this point into the shrunk population.

Step 7: Set $n_s = n_s + 1$, go to step 8.

Step 8: Check whether all points in current population are examined: if yes, go to Step 10; otherwise, go to Step 9.

Step 9: Set $N = N + 1$, and repeat Step 3 to Step 7 for the Nth point.

Step 10: Form the shrunk population with points obtained from the previous steps.

Step 11: Examine whether there is any point in the shrunk population: If yes, go to Step 12; otherwise, end the procedure (i.e., all the points with a non-zero prediction interval are to be evaluated using simulation).

Step 12: Sort the points in the shrunk population by the value of the prediction interval (from large to small) to obtain a sorted shrunk population.

Step 13: Examine the N_sth point with the criterion described in Eq. (3.2): If satisfied, it means this point as well as remaining points in the sorted shrunk population can be estimated using kriging metamodels; otherwise, go to Step 15.

Step 14: Form the set of points that can be estimated using kriging metamodels.

Step 15: Evaluate the N_sth point using simulation.

Step 16: Check whether all the points in the sorted shrunk population have been examined: If yes, end the procedure; otherwise, go to Step 17.

Step 17: Set $N_s = N_s + 1$, and go back to Step 12.

This procedure is repeated for all objective functions in the simulation. If a point is selected to be evaluated using kriging for all objective functions, then that point is estimated using kriging; otherwise, that point is evaluated using the simulation.

3.3.4 Stopping Criterion

In order to compare the performance of the proposed approach with a conventional MOGA, a stopping criterion is defined as follows. This stopping criterion has two parts, and both parts need to be satisfied to stop the optimization procedure:

1) When the number of non-dominated points is more than some pre-specified percentage of the population size (e.g., 80%) and becomes steady, it can be concluded that the algorithm has converged or is near the Pareto set.
2) When the iteration history, i.e., the curve representing the number of function calls versus the number of generations becomes flat, it can be concluded that the algorithm has been converged.

Note that the second part of the stopping criterion is not used for the conventional MOGA while the first part is. This is because in the conventional MOGA the simulation is used for any newly generated points in each generation and thus the number of function calls always increases as the generation number increases (see Figure 4.12).

3.3.5 Steps for the Kriging Metamodeling Assisted MOGA (or K-MOGA)

In the proposed approach, the points in a population, other than those in the initial population, can be divided into two groups. Those points whose simulation response can be estimated using kriging metamodels and the remaining points whose response are to be evaluated using the simulation. Again, the fitness values for all points in the initial population are calculated by running the simulation. Kriging metamodels are constructed and updated based on all function calls. Since the initial points may be far away from the Pareto frontier or they may not sample the design space well, the initial kriging metamodels may not be sufficiently accurate. However, these kriging metamodels are adaptively improved as the algorithm evolves and more function calls are used. During the early generations, the percentage of points for which the kriging metamodels are used is small (e.g., this percentage is zero for the initial population). However, as more observed points are added to the kriging metamodel, the predicted error of the unobserved points is going to gradually improve and the percentage of the points for which kriging metamodels is used is going to increase as the subsequent generations are evolved.

Note that, according to the criteria in Eq. (3.4), points with large variance values are required to be evaluated using the simulation and such points will be used to improve the accuracy of the kriging metamodels. Actually, all evaluated points contribute to improve the accuracy of the kriging metamodels, no matter whether they are in the current population or previous populations. Thus, the kriging metamodel accuracy will always improve during the optimization procedure.

Figure 3.5 shows a one generation scheme for the proposed kriging assisted fitness evaluation approach in online approximation. A conventional MOGA (see, e.g., Section 2.4.1) is used as the population-based optimization framework, as shown in the dashed block on the left hand side in Figure 3.5. Compared to the conventional MOGA, there is an addition in the proposed approach. That is, there is an online metamodeling assisted fitness evaluation, as shown in the dashed block on the right part in Figure 3.5. We denote this MOGA with kriging metamodeling assisted fitness evaluation as K-MOGA.

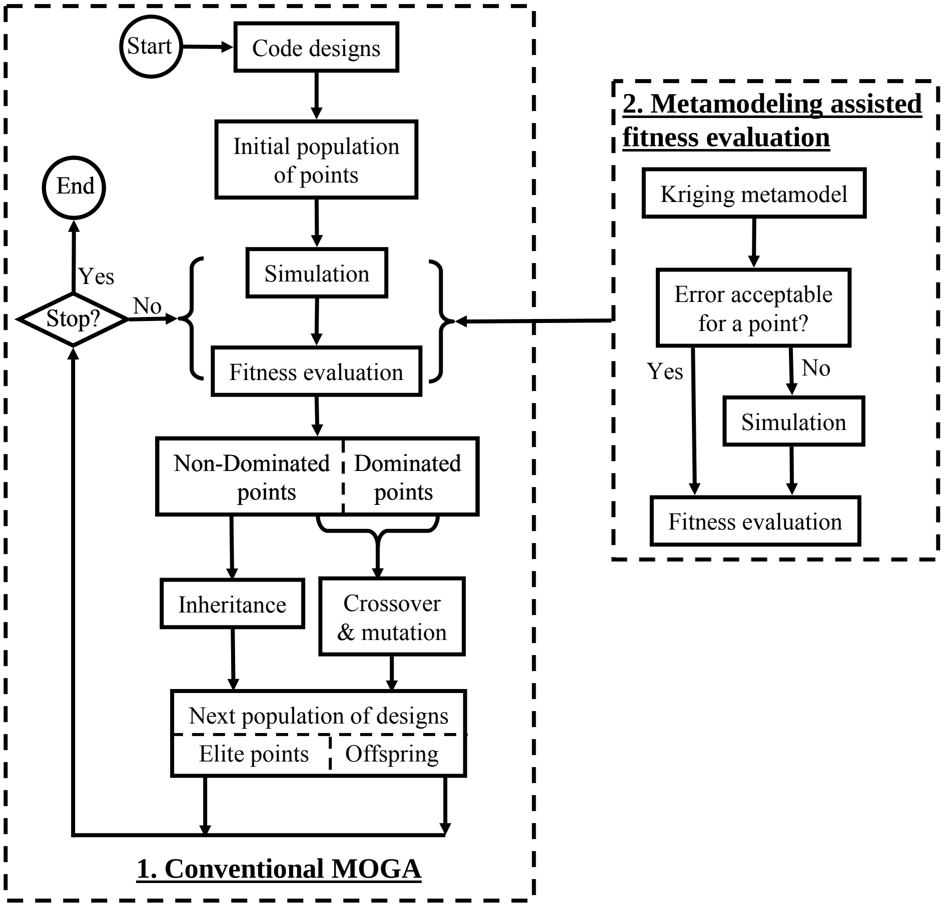


Figure 3.5 Flowchart of K-MOGA in one generation

The steps for the proposed approach are as follows:

Step 1: Start with generating an initial population. Simulation is used to calculate the responses (i.e., objective/constraint functions) for all points in the initial population and these are used to build the initial kriging metamodels, each for an objective/constraint function. The non-dominated (or elite) points in the previous population are identified based on their fitness values.

Step 2: The algorithm evolves into the next generation. The non-dominated (or elite) points in the previous population are inherited into the next generation. The remaining (but not the inherited) points are generated by the crossover and mutation.

Step 3: Apply the current kriging metamodels to predict response values and the associated variance for the points in the current population; identify the estimated non-dominated and estimated dominated points and calculate *MMD* for the current population.

Step 4: Obtain the prediction interval for each point and apply the sort, shrank, and select algorithm to split the current population into two groups: Those points that can be estimated using kriging metamodels and those that need to be evaluated using simulation.

Step 5: Calculate the Fitness value of each point based on its estimate (by kriging metamodels) or evaluation (by simulations), and obtain the non-dominated points in the current population. Note that function calls are saved in this step, since some points are estimated using metamodels instead of simulation. Kriging metamodels are also updated based on the newly available function calls.

Step 6: Check the stopping criterion. Check the stopping criterion described in Section 3.3.4. If both stopping parts in the stopping criterion are satisfied, stop the procedure; otherwise, go back to Step 2.

In the next section, several numerical and engineering examples are used to demonstrate the applicability of the proposed approach. Discussions are also carried out based on the obtained results.

3.4 Examples and Results

In this section, we use five numerical and engineering examples with different degrees of difficulty to illustrate the applicability of the proposed approach. As a typical example of our results, we use the first example, a simplified version of ZDT2 (e.g., Deb, 2001), to present a comparison of the conventional MOGA and the proposed approach. Next, the results for a real-world engineering example (Rolander et al., 2005; Ramb and Joshi, 2005) are presented. Finally the comparison results for the remaining three examples selected from the literature (Deb, 2001) are presented.

In order to compare the conventional MOGA and the proposed approach, both approaches were run for 30 times for each example. The values of other genetic parameters are selected as follows. For the convergence, the number of non-dominated points is limited to be not more than a pre-specified percentage of the population size (i.e., 70% of the population). The remaining points are offspring points that are produced by genetic operations such as crossover and mutation. Such a strategy ensures that a pre-specified percentage of points in the population (i.e., 30%) are generated by genetic operations. For offspring points, we use a probability of 0.95 for crossover and a probability of 0.05 for mutation. The same settings are used for all examples.

Note that the proposed kriging metamodeling assisted fitness evaluation approach is implemented in a MOGA framework for the rest problems of this chapter. Thus, as defined in Section 3.3.5, we use “K-MOGA” to denote the proposed approach and use “MOGA” to denote the conventional MOGA in the figures and tables in the rest of the chapter.

3.4.1 ZDT2 Example

We applied the conventional MOGA and the proposed approach to a simplified ZDT2 problem (Deb, 2001). This example has two objective functions, no constraint, and three variables, as given in Eq. (3.6).

$$
\begin{aligned}
\text{minimize} \quad & f_1(x) = x_1 \\
& f_2(x) = g(x) \times h(x) \\
\text{where} \quad & g(x) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i \\
& h(x) = 1 - \left(f_1(x)/g(x)\right)^2 \\
& n = 3 \\
& 0 \le x_i \le 1, \quad i = 1, \dots, n
\end{aligned}
\tag{3.6}
$$

The true Pareto optimal solutions for this problem are: x_1 being any point within the range [0, 1], $x_2 = 0$, and $x_3 = 0$, with the true Pareto frontier as shown in Eq.(3.7).

$$
\begin{aligned}
& f_2 = 1 - f_1^{\,2} \\
& 0 \le f_1 \le 1
\end{aligned}
\tag{3.7}
$$

The Pareto frontier is non-convex as shown in Figure 3.6. For this example, two separate kriging metamodels (for the two objectives) are built and adaptively improved to predict the response for the objective functions values. It can be seen that the results from K-MOGA are in good agreement with MOGA and the true Pareto frontier.

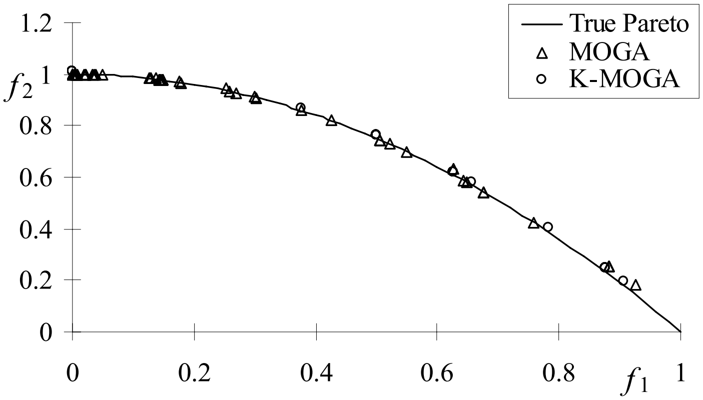


Figure 3.6 Pareto solutions for the ZDT2 example

Figure 3.7 shows the number of function calls for 30 different runs (to account for the stochastic property of MOGA). As shown in this figure, a MOGA run with the least number of function calls (i.e., 231 in run 15) requires more calls than the new approach run with the maximum number of function calls (i.e., 197). The mean value and standard deviation (STD) for all 30 runs for both the conventional MOGA and new approach are shown in Table 3.1.The performance of the conventional MOGA and the proposed K-MOGA approach for all examples in terms of function calls are presented in Table 3.1. As shown in that table, on the average, the K-MOGA can save over 50% in terms of the number of function calls when compared to the conventional MOGA. For most of the test examples, standard deviations in the proposed approach are also less than those in the conventional MOGA.

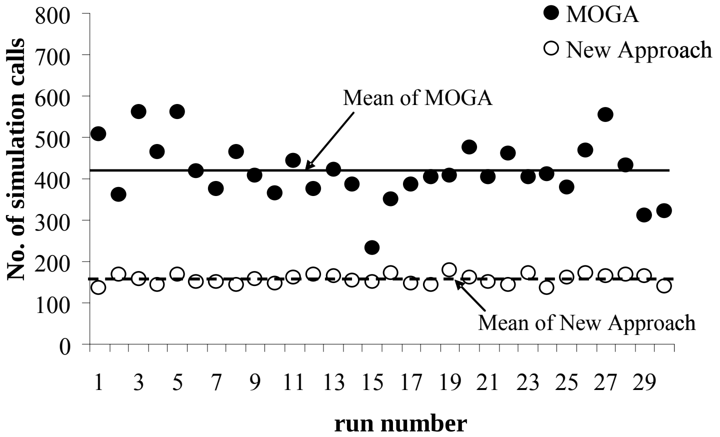


Figure 3.7 Number of function calls vs. run no. for the ZDT2 example

The results for the ZDT2 example show that the proposed approach saves more than 60% of the number of function calls on the average compared to the conventional MOGA.

3.4.2 Cabinet Example

The optimization of a fully enclosed vertical cabinet containing ten point blade server racks is selected as the last engineering test example. The simulation model for this example was developed by Rolander et al. (2005). The cabinet geometry is shown in Figure 3.8.

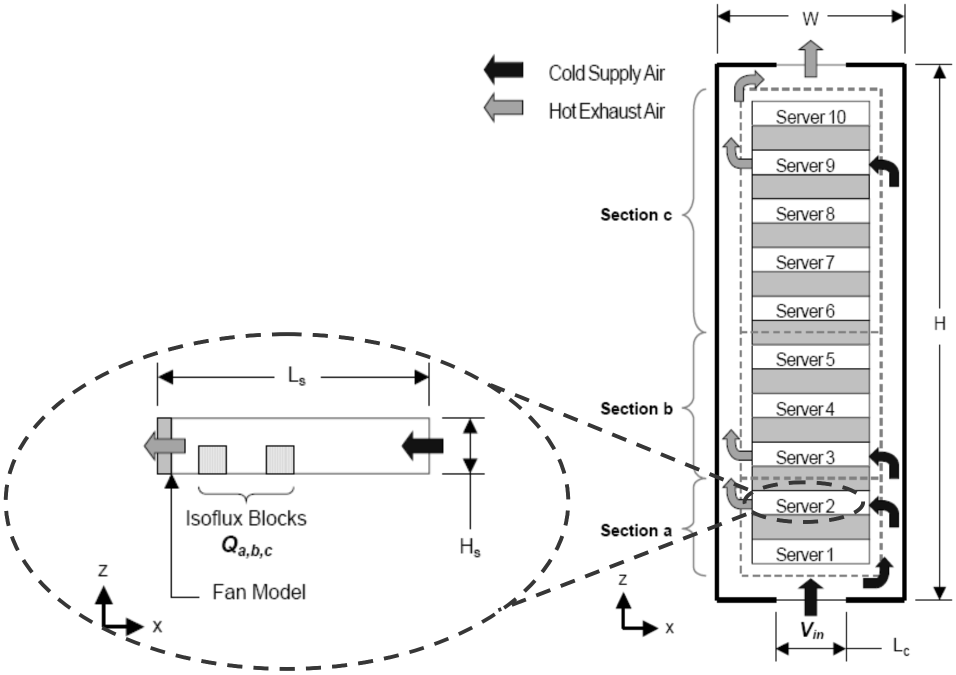


Figure 3.8 Air-cooled cabinet model design, courtesy of Rolander et al. (2005)

This model has two isoflux blocks that act as flow obstructions, representing a dual processor blade server. Both blocks have a constant heat generation rate Q, which is dissipated through convection to the air flowing through the server. The cabinet is divided into three sections: a, b and c, corresponding to the lower two, middle three, and upper five servers as shown in Figure 3.8. The quantities Q_a, Q_b, and Q_c denote the heat generation of each server in the respective cabinet section. In each section, all blocks have the same constant heat generation rate Q. Air flow with an inlet velocity is used to cool the cabinet as shown in Figure 3.8. A two-dimensional heat transfer simulation model has already been built for this problem and reported in the literature (Rolander et al., 2005; Ramb and Joshi, 2005).

In this thermal optimization model, inlet air velocity V_{in} and heat generation Q_{a}, Q_{b}, and Q_{c} are considered as design variables. The output of the simulation model is a 10-element vector of temperatures, each for one server. The design objectives are to: 1) minimize the maximum server temperatures, and 2) maximize the sum of total heat generations. The optimization problem is defined in Eq. (3.8). The constraint in this optimization problem is handled using a metamodeling assisted constraint method introduced in Li et al. (2006b).

$$
\begin{aligned}
&\underset{Vin,\, Q_{a,b,c}}{\text{Minimize}} && f_1 = Max\ (T_j)\quad j = 1,\dots,10 \\
&\underset{Vin,\, Q_{a,b,c}}{\text{Maximize}} && f_2 = 2Q_{\mathrm{a}} + 3Q_{\mathrm{b}} + 5Q_{\mathrm{c}} \\
&\text{subject to} && \\
& && max\ (T_j)/85 - 1 \le 0 \quad j = 1,\dots,10 \\
& && 0.2 \le Vin \le 2 \\
& && 2 \le Q_{\mathrm{a,\,b,\,c}} \le 200
\end{aligned}
\qquad (3.8)
$$

The optimization results for this cabinet problem are shown in Figure 3.9 and in Table 3.1. In this problem, each function call generates the sum of Q and the maximum of T_j, which are used in both objective and constraint functions. It is observed that the number of function calls in the proposed approach is significantly fewer than the conventional MOGA, while the optimal solutions as shown in Figure 3.9 are comparable.

The results in Figure 3.9 also show several interesting and unexpected features. It is seen that the maximum total power dissipation with this example air-cooled geometry is limited to about 1800W, beyond which the maximum temperature specification cannot be met. It is extremely useful to deduce such limits for various cooling solutions, so that one could determine the overall thermal management approach for a given total cabinet power dissipation. Also, due to the rather flat shape of the Pareto Front in Figure 3.9 it is seen that proper placement of servers at the higher end of total powers is very important in reducing temperatures.

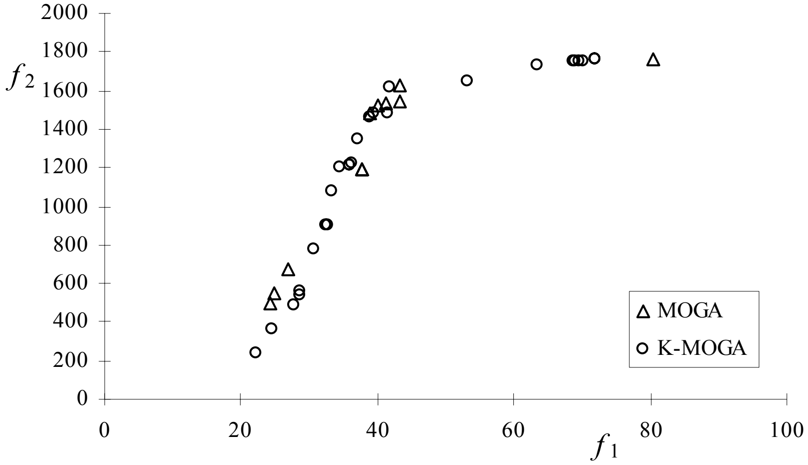


Figure 3.9 Pareto solutions for cabinet problem using MOGA and K-MOGA

3.3.6 Additional Examples

In this section, three additional examples: ZDT1, ZDT3, and Gear-Train from Deb (2001) are presented to demonstrate further applicability of the proposed approach. These three test problems have different degrees of difficulty and characteristics. For instance, the Pareto frontier for ZDT3 is disconnected. And, the Gear Train test example has integer design variables. The formulations of these test examples and the comparison results as obtained from the proposed approach, the conventional MOGA, and the true Pareto frontier if the closed form solution is available, are presented as follows: formulations of the examples are given in Eq. (3.9), Eq. (3.10) and Eq. (3.11); Pareto frontiers are given in Figure 3.10, Figure 3.11 and Figure 3.12.

ZDT1 Example:

$$\begin{aligned}
\text{minimize} \quad & f_1(x) = x_1 \\
& f_2(x) = g(x) \times h(x) \\
\text{where} \quad & \\
& g(x) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i \\
& h(x) = 1 - \sqrt{f_1(x)/g(x)} \\
& n = 3 \\
& 0 \le x_i \le 1, \quad i = 1, \dots, n
\end{aligned} \tag{3.9}$$

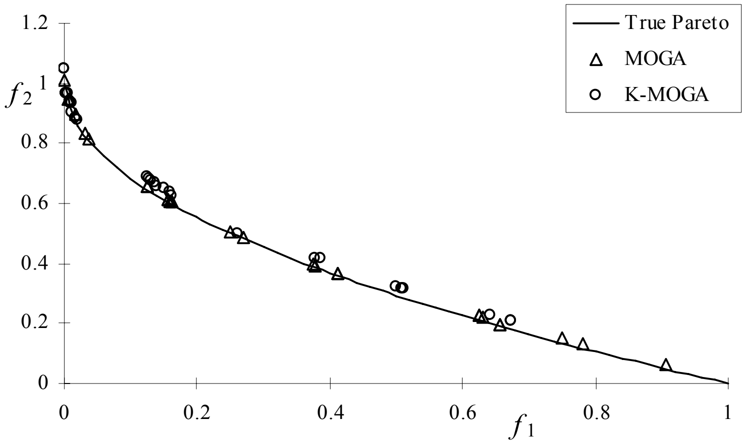


Figure 3.10 Pareto frontiers for ZDT1 example

ZDT3 Example:

$$\begin{aligned}
\text{minimize} \quad & f_1(x) = x_1 \\
& f_2(x) = g(x) \times h(x) \\
\text{where} \quad & \\
& g(x) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i \\
& h(x) = 1 - \sqrt{f_1(x)/g(x)} \\
& \qquad - \left(f_1(x)/g(x)\right)\sin(10\pi f_1) \\
& n = 3 \\
& 0 \le x_i \le 1, \quad i = 1, \dots, n
\end{aligned} \tag{3.10}$$

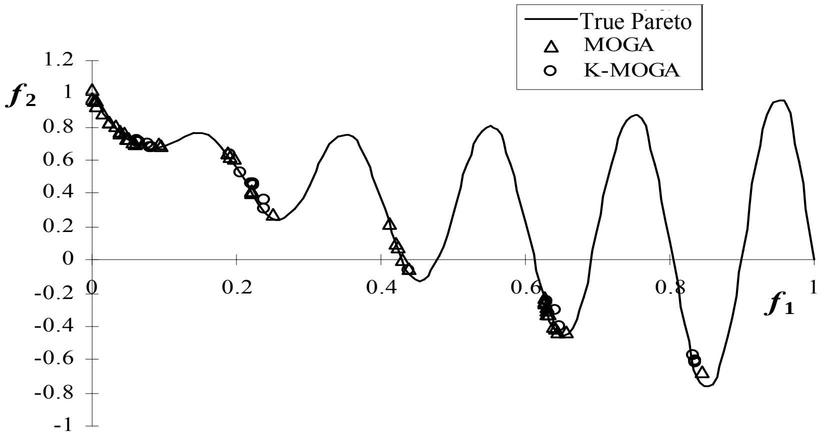


Figure 3.11 Pareto frontiers for ZDT3 example

<u>Gear-Train Example</u>:

$$\begin{aligned}
\text{minimize} \quad & f_1(x) = \log_{10}\left(\left(\frac{1}{6.931} - \frac{x_1 x_2}{x_3 x_4}\right)^2\right) \\
& f_2(x) = \max\,(x_1, x_2, x_3, x_4) \\
\text{where} \quad & \\
& 12 \le x_1, x_2, x_3, x_4 \le 60 \\
& x_i \text{ are integers}
\end{aligned} \qquad (3.11)$$

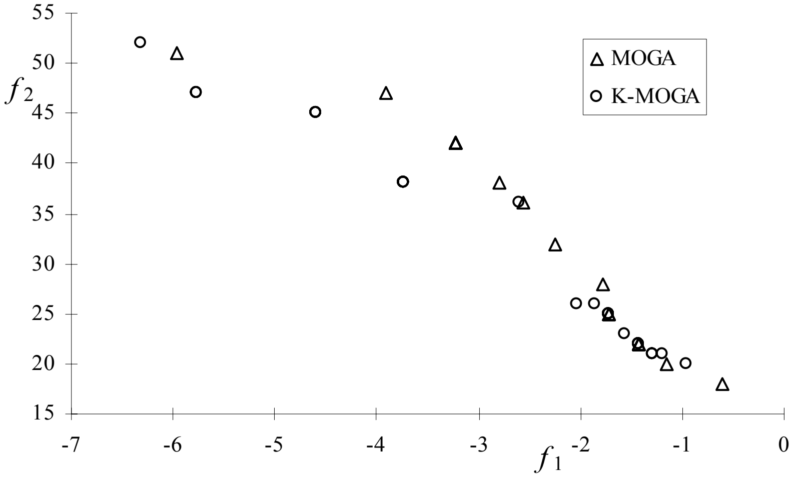


Figure 3.12 Pareto frontiers for Gear-Train example

The performance of the conventional MOGA and the proposed K-MOGA approach for all examples in terms of function calls are presented in Table 3.1. As shown in that table, on the average, the K-MOGA can save over 50% in terms of the number of function calls when compared to the conventional MOGA. For most of the test examples, standard deviations in the proposed approach are also less than those in the conventional MOGA.

Table 3.1 Comparison of MOGA and Proposed approach for test examples

Example	Pop-size	% average reduction in # of function calls	Number of function calls					
			Conventional MOGA			K-MOGA		
			30 runs	Mean	STD	30 runs	Mean	STD
ZDT1	30	57%	[293-594]	398	51	[120-258]	172	38
ZDT2	30	63%	[231-561]	417	73	[134-197]	156	12
ZDT3	30	50%	[340-616]	485	69	[176-307]	243	34
Gear-Train	50	50%	[347-633]	471	78	[159-323]	234	39
Cabinet	30	42%	[810-1125]	998	75	[434-705]	584	75

3.4 Summary

A new kriging metamodeling assisted fitness evaluation approach in online approximation is presented in this chapter. In the proposed approach, the kriging metamodeling is embedded within a conventional MOGA. Compared to a conventional MOGA, the proposed approach reduces the number of function calls by evaluating some points in the population via kriging metamodels instead of the simulation. We have introduced the concept of “minimum of the minimum distances” (*MMD*) and derived its relation with the predicted error that is easily obtained from kriging. This criterion is used to identify those points in the population that can be evaluated using kriging metamodels. The identified points are those that do not change the estimated dominance status in the objective space for the current generation. For other points in the generation, the responses are obtained from the simulation and used to adaptively update the next generation kriging metamodels so that more points can be evaluated by the updated kriging metamodels and thus an additional number of function calls can be saved in subsequent generations.

In the proposed approach, the metamodel may be of low fidelity and it may even produce false optima, which is one of the general concerns in using a metamodel, can be avoided. The proposed criterion is objective rather than subjective and can be applied to other population-based optimization methods using different types of metamodels if the measure for predicted error is available. The main advantage of using kriging metamodeling is that the predicted error of the estimated response can be obtained without extra computational effort.

Five examples of both numerical and engineering types and with different degrees of difficulty are used to demonstrate the applicability of the proposed approach. The results show that the proposed is able to achieve comparable convergence and diversity of the Pareto frontier as to that from a conventional MOGA while at the same time to significantly reduce the number of function calls.

One of the characteristics of the proposed approach is that those points inherited from previous generations with incorrectly estimated kriging variance are most likely to be removed from the population by a more accurate kriging metamodel. Therefore, the side effect of such inherited points can be diminished when the kriging metamodels are updated adaptively in consecutive generations. In essence, the proposed approach has a self-correcting mechanism in terms of identifying proper points for kriging metamodeling to estimate.

Finally, it should be noticed that the relation between *MMD* and the predicted error of kriging for the objective functions is devised based on a worst case scenario and thus the proposed approach can be considered to be conservative. Also, the points that are generated by GA operations, evaluated by the simulation and used in kriging metamodel construction can affect the accuracy of the metamodel. By devising a less conservative criterion and an improved sampling strategy for kriging, it should be possible to further improve the efficiency of the proposed approach in terms of the number of function calls.

In the next chapter, the research thrust two, a new adaptive DOE method in online metamodeling, is presented.

Chapter 4 Adaptive DOE in Online Metamodeling

4.1 Introduction

As discussed in Chapter 3, fitness estimation in online metamodeling is a popular and efficient strategy for reducing the number of function calls in the optimization process. However, in using such a strategy, the number of function calls can only be reduced during each generation. Another way of saving the number of function calls is to reduce the total number of generations required for convergence. Many approaches were developed in the literature that use metamodeling for generating some new design points (Anderson and Hsu, 1999; Abboud and Schoenauer, 2001; Rasheed, 2000; Rasheed et al., 2005; Li et al., 2007) to speed up the convergence. Such approaches approaches actually fall under DOE methods, since new points are selected according to some predefined criterion.

The use of metamodeling for fitness evaluation can increase the risk of generating false optima (Jin, 2005), especially when the metamodel is not accurate in the early generations of a population based optimization method (e.g., MOGA in this dissertation). This risk of generating false optima is avoided when the metamodel is used during the generation or reproduction of new design points (Shan and Wang, 2005). However, the use of metamodeling during reproduction cannot reduce as many of the function calls as it does in the fitness evaluation. Moreover, all previous metamodeling methods for reproduction modify the operators in a population based optimization method (e.g., crossover and mutation in MOGA) and thus lack the ability to directly estimate the Pareto frontier. Finally, the savings obtained in the number of function calls using metamodeling for fitness evaluation and reproduction are independent of each other, since such savings are obtained during different stages of a generation in a population based optimization method (e.g., MOGA). For this reason, an integration of metamodeling with both fitness evaluation and reproduction should reduce the number of function calls significantly more than previous population based optimization methods. Such an approach has not been reported in the literature and is the subject of the present chapter.

The population based optimization method used in this chapter is a conventional MOGA (recall Section 2.4.1), thus the overall approach with the DOE and fitness estimation is denoted as an improved MOGA in the rest of this chapter. In the improved MOGA, the approach discussed in Chapter 3 is used for determining whether metamodeling should be used for fitness evaluation for some of the design points. The improved MOGA also involves a newly developed, adaptive DOE approach, with a better sampling of the design space, which creates a portion of the design points for the MOGA.

This proposed improved MOGA has several advantages over previously reported metamodeling-assisted MOGA approaches. First, in our method the metamodeling is used only to estimate the response for design points whose response error is small. Previous methods, however, use metamodeling for points whose response error might be large. Thus, in our method, the average metamodel accuracy is not as important as it is for previous online metamodeling methods. Second, our improved MOGA uses metamodeling for both fitness evaluation and reproduction, which helps to reduce the number of function calls significantly. This is because the use of metamodeling in the fitness evaluation reduces the number of function calls in each generation of a MOGA, while its use in reproduction reduces the total number of generations. Finally, the improved MOGA uses objective measures for using metamodeling in the fitness evaluation and reproduction, while previous approaches use either subjective measures or do not have a predictive ability for the Pareto frontier.

The rest of the chapter is organized as follows. The overall framework for the improved MOGA is introduced in Section 4.2. Details of the proposed adaptive DOE method are given in Section 4.3. In Section 4.4, the stopping criterion and the steps for the improved MOGA are outlined. In Section 4.5, the results of numerical and engineering examples are presented. Key conclusions are provided in Section 4.6.

A portion of this chapter was presented in Li et al. (2007).

4.2 Framework for the Improved MOGA

Compared to the conventional MOGA, there are two additions in the proposed improved MOGA. These additions are (i) online metamodeling-assisted fitness evaluation, and (ii) adaptive DOE-assisted reproduction. The first addition has been presented in Chapter 3. However, the focus of this chapter is on a newly developed, adaptive DOE technique that is used as part of a scheme for reproduction of new design points. Another focus is on the integration of the two additions within the conventional MOGA.

Figure 4.1 illustrates the three main components in the improved MOGA, as shown inside the dashed blocks of Figure 1 and labeled as components 1, 2, and 3. These components are: 1. Conventional MOGA, 2. Metamodeling-assisted fitness evaluation, and 3. Adaptive DOE-assisted reproduction. Component 1 essentially summarizes the basic steps in a conventional MOGA (recall Section 2.4.1). Component 2: Metamodeling assisted fitness evaluation, outlines the scheme for fitness evaluation when metamodeling used, as presented in Chapter 3.

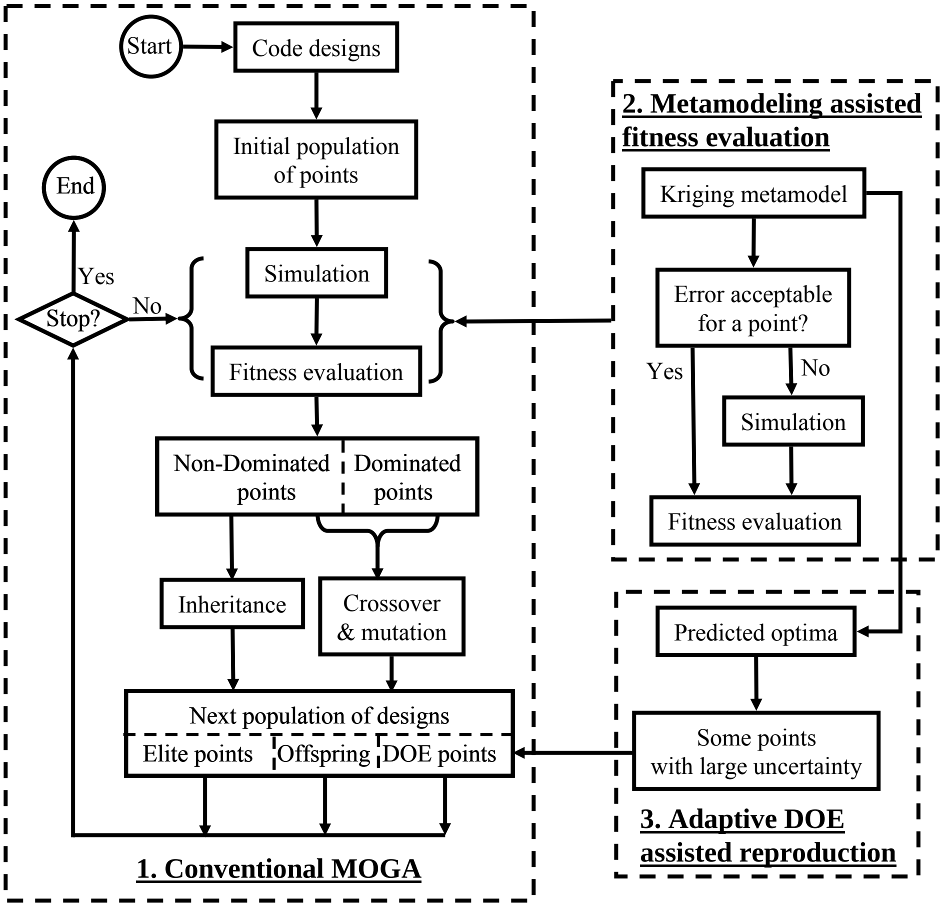


Figure 4.1 The framework for the improved MOGA

Component 3, the adaptive DOE-assisted reproduction, shows a new adaptive DOE method that is proposed for generating a portion of the design points used in the next population. Note that as shown in Figure 4.1, the kriging metamodels constructed and updated by Component 2 are also used to obtain the predicted optima. Some unobserved design points in the predicted optima are then selected and added to the next population as the DOE points. This new DOE method is described next in Section 4.3.

4.3 Adaptive DOE Assisted Reproduction

As discussed before, the DOE points are selected from the predicted optima. The points in the predicted optima are obtained by applying the conventional MOGA to an optimization model in which the objective and constraint functions are replaced by their kriging metamodels. Some of the points in the predicted optima might have been previously generated as elite points, offspring points, or DOE points. Moreover, a subset of these points might have been observed in the previous generations.

The DOE points selected from the predicted optima are those that have a large estimated error. These DOE points are observed, and the actual values of their responses are used to update the kriging metamodel. In this way the estimated error of other unobserved design points around the selected points can be reduced. In addition, the predicted optima can eventually become a good approximation to the Pareto frontier. The approach has an adaptive ability: the predicted optima gradually and adaptively approach the Pareto frontier as the MOGA converges.

Next, we must determine which points and how many of the points from the predicted optima should be selected. In the next sub-section, a constrained maximum entropy design method is proposed to solve this selection problem.

4.3.1. DOE Points: Selection by Constrained Maximum Entropy Design

The DOE points are selected by a constrained maximum entropy design approach. This approach is an extension to the original unconstrained maximum entropy design approach (recall Section 2.6).

An important property of the original unconstrained maximum entropy design method is that the newly selected design point is located far from all the observed design points. This property is exactly what we are looking for in selecting the DOE points from the predicted optima. In addition, we still need a threshold for determining whether the selected DOE points have a large estimated error. A design point is considered to have a large estimated error if this error can change the domination status of this point. For example, as shown in Figure 4.2, the corresponding estimated error intervals for the two design points *a* and *b,* with a presumed confidence level, have an overlap along the objective function f_1 axis.

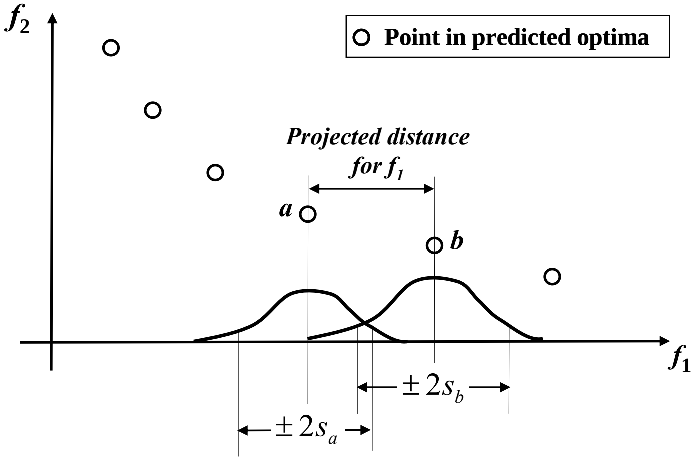


Figure 4.2 Determination of a large estimated error

Due to this overlap, there is a possibility that one of these points can become a dominated point even though it belongs to the predicted optima. Thus, either point *a* or point *b* or both could be selected as DOE points. Figure 4.3 represents a situation in which point *a* is dominated by point *b*, when both of their response estimates vary within the error intervals. The solid points in the figure are used to represent the response estimates due to uncertain variations. In this situation, as shown in Figure 4.3 the domination status of point *a* has been changed (compared to Figure 4.2) and thus the point will have to be evaluated (or observed).

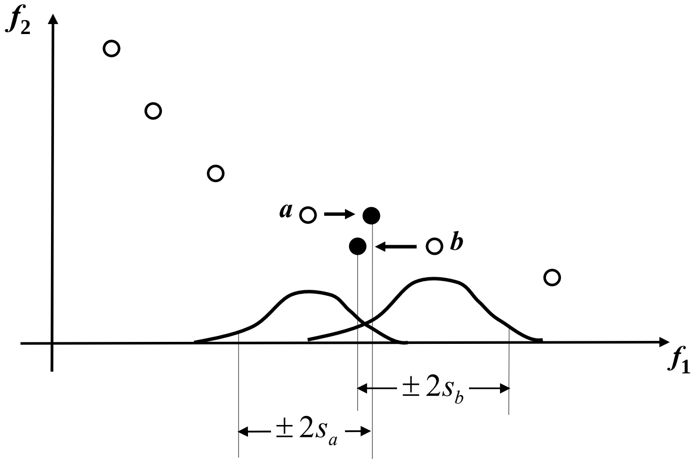


Figure 4.3 Change of the determination status

The condition that determines whether a point has a large estimated error is stated, in general, as follows:

$$\frac{1}{2}\left(I_p(a)+I_p(b)\right) > \text{projected distance for } f_m \tag{4.1}$$

where *a* and *b* are the two adjacent points in the current predicted optima along at least one f_m with $i \in \{1, 2, ..., M\}$, $I_p(a)$ denotes the predicted interval (Eq. (2.12)) for the estimated response for the point *a*. Therefore, a new constrained maximum entropy design approach is proposed in which the DOE points are selected from the predicted optima one point at a time so that:

$$\begin{aligned} & x_{i+1} = \arg\max \det(R), x_{i+1} \in U \\ & \text{subject to } \frac{1}{2}\left(I_e(x_{i+1}) + I_e(x')\right) > \text{projected distance for } f_m,\ m \in \{1, 2, ..., M\} \end{aligned} \tag{4.2}$$

where *U* is the set of unobserved points in the current predicted optima, *R* and *i* follow the same definition as in Section 2.6, and x' is an adjacent point to x_{i+1} in the current predicted optima. Note that if x' is an observed point and the constraint in Eq. (4.2) is violated, then the next unobserved point in the predicted optima will be used as x'.

4.3.2. DOE Points: Selection Steps

The steps in the DOE approach, which is an adaptive DOE assisted reproduction approach, are presented in the flowchart of Figure 4.4.

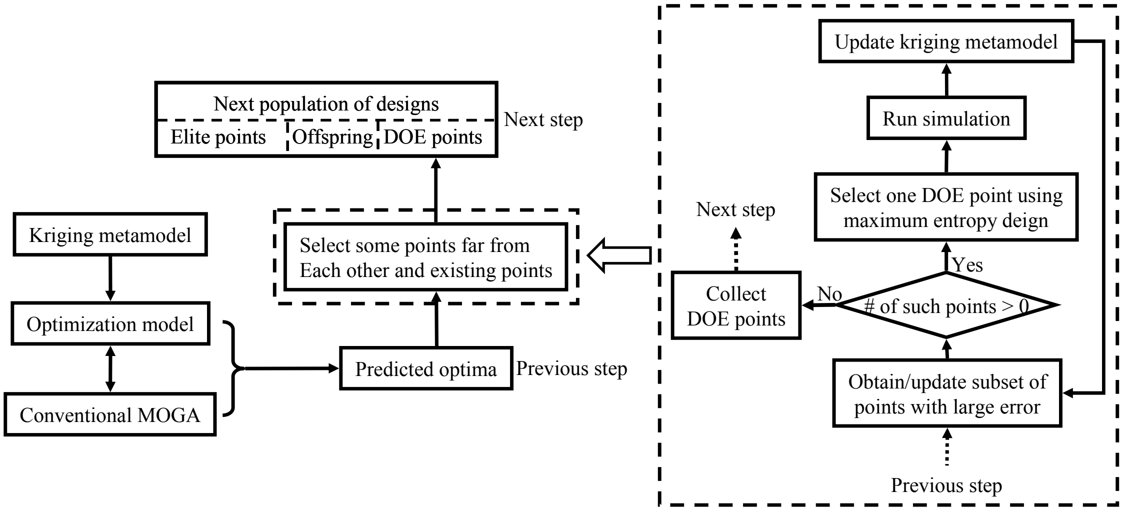


Figure 4.4 Steps for adaptive DOE assisted reproduction

As shown in Figure 4.4, from the left, beginning from the kriging metamodel, the optimization model is constructed such that all objective and constraint functions are replaced by their kriging metamodels. Next, the conventional MOGA is used to solve this optimization model. The result is a set of optimum solutions, called predicted optima. Next, the constrained maximum entropy design method is used to select some points from the predicted optima that are far from each other and existing design points (as shown in the dashed box in Figure 4.4), which are then augmented as the DOE points to the next population.

The steps inside the dashed block to the right of Figure 4.4 work as follows. First, based on the current kriging metamodel, we identify a subset of points from the predicted optima that have a large error (see Eq. (4.2)). If this is the first iteration for the steps in the dashed block, we call the obtained subset of points the “first iteration subset.” We call subsequent iterations “updated subsets.” Next, we check the stopping criterion to see whether this set is empty. There can be two situations when the set will be empty. One situation is when the first iteration subset is empty, which means that all points in the predicted optima have a small error. In this case, we go to the “Next step” outside the dashed block without a DOE point. The other situation is when an updated subset is empty after several iterations within the dashed block. In this case, in every iteration before the current iteration, a DOE point has already been selected, and all of these points are collected and sent to the “Next step.” Note that all of the sampled (or DOE) points are evaluated one at a time using the simulation, and the kriging metamodel is updated accordingly. In this way, it is likely that the points which previously had a large error will have their error reduced once the kriging metamodel has been updated. As a result, except for the one DOE point that is selected in the last iteration, an updated subset can have fewer points than the last updated subset. Eventually the updated subset becomes empty, which ends the steps inside the dashed block.

4.4 Improved MOGA

In this section, the steps for the improved MOGA are given. The stopping criterion used in the improved MOGA is similar to that in K-MOGA of Chapter 3, except that when the number of non-dominated points is reached to more than 70% (it was 80% in K-MOGA), it can be concluded that the algorithm has converged. The change of this percentage is due to the fact that the new points are generated by DOE, inheritance, crossover and mutation in the approach in this chapter, while there was no DOE in the approach in Chapter 3.

4.4.1 Overall Procedure for the Improved MOGA

The detailed steps for the proposed improved MOGA approach are given as follows, including the initial and main steps.

Initialization Step: The design points are coded in a binary form. An initial population of design points is created randomly. Simulation is used to calculate the responses (i.e., for the objective and constraint functions) for all design points in the initial population, and these points are used to build initial kriging metamodels, each for an objective/constraint function.

Main Steps:

Step 1: Kriging metamodel updating and fitness evaluation. The kriging metamodel is updated for the current population of design points. The responses for design points that have an acceptable estimated error in their response are evaluated/updated by the kriging metamodels. The remaining design points in the population are evaluated by the simulation. The fitness of all design points is evaluated using a non-dominated sorting algorithm (Deb, 2001).

Step 2: Non-dominated/dominated design points and the GA operations. The non-dominated and dominated design points are determined. The non-dominated design points are inherited by the next population to form the elite points. The GA operations of crossover and mutation are applied to non-dominated and dominated design points to create the offspring design points for the next population.

Step 3: DOE points. Based on the current kriging metamodels, the adaptive DOE-assisted reproduction is performed to obtain the DOE points (recall Figure 4.1 and Figure 4.4). DOE points are augmented to the next population. Thus, the next population contains design points generated by MOGA operations as well as the DOE points.

Step 4: Stopping criterion. If both parts (recall Section 3.3.4) of the stopping criterion are satisfied, the algorithm is stopped; otherwise, the algorithm continues to Step 1.

4.5 Examples and Results

Five examples are used to demonstrate the proposed approach. The first example is an engineering example (i.e., the distillation column design example) with a simulation model for which explicit (or closed form) functions are not available. The other four examples are numerical or engineering type with explicit objective functions, which were also been used in Chapter 3. The conventional MOGA, the K-MOGA, and the improved MOGA are applied to these examples.

For convergence of the conventional MOGA, the number of non-dominated design points inherited is set to be 70% of the population. We use a probability of 0.95 for crossover and 0.05 for mutation. For the improved MOGA, the total number of the inherited and DOE points do not exceed more than 80% of the population. Likewise, K-MOGA, the total number of the inherited and DOE points does not exceed more than 80% of the population. This is to ensure that the percentage of the design points from crossover and mutation is not less than 20%, so that the stochastic property of MOGA is preserved.

4.5.1 Distillation Column Design Example

A distillation column (or tower) is a unit used to separate a mixture of two or more chemicals based on the difference in their volatilities. A layout for a binary distillation tower (for separating a mixture of two chemicals) is given in Figure 4.5.

The symbols used in Figure 4.5 are defined as follows. D and B are the flow rate of distillate and bottom products; F is the feed flow rate; L is the liquid molar flow rate inside the column; N_1 is the number of stages in the rectifying section (upper section); N_2 is the number of stages in the stripping section (bottom section); R is the reflux ratio; V is the vapor molar flow rate inside the column; V_B is the boil up ratio; X_B is the composition of the more volatile component in stream B; X_D is the composition of the more volatile component in stream D; X_F is the mole fraction of the more volatile component in the feed; X_N is the mole fraction of more volatile component in liquid leaving tray N; and Y_N is the mole fraction of more volatile component in vapor leaving tray N.

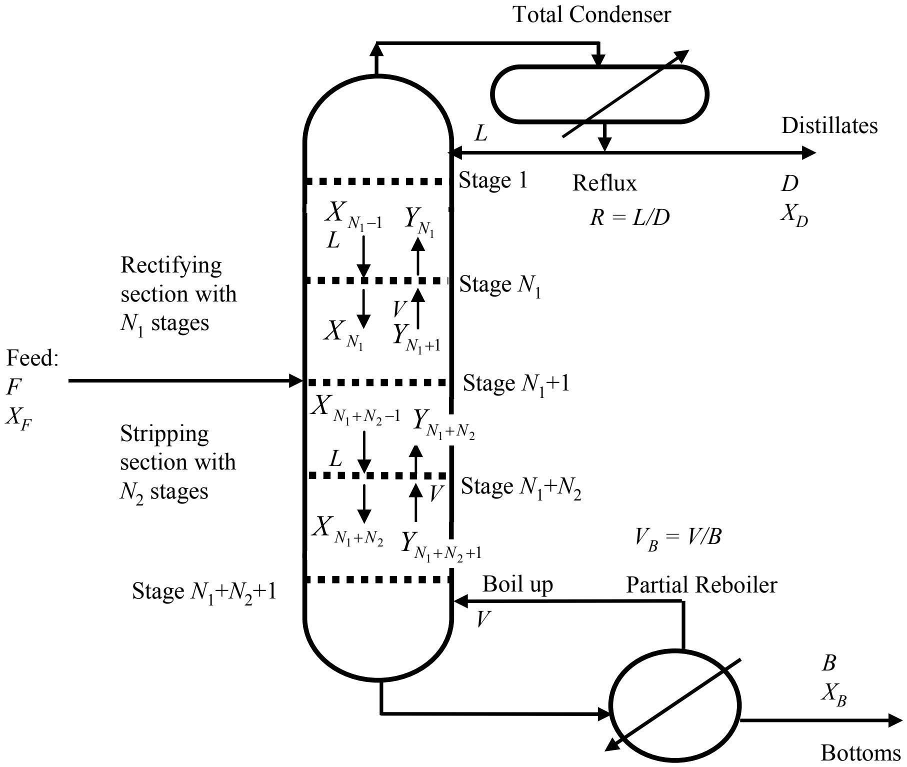


Figure 4.5 Binary distillation column

In this example a mixture of toluene (less volatile component) and benzene (more volatile component) is fed to a distillation column where they are separated into two high purity streams. The stream recovered at the top is the distillate and would contain mostly the more volatile component benzene. Similarly, the liquid recovered at the bottom is called the bottoms and would contain mostly the less volatile component, toluene. A detailed description of the distillation towers can be found in Jeankoplis (1993) as well as Seader and Henley (2006).

The attained purity of the distillate and bottoms depend on many design variables. In this simplified version of the distillation tower, the design variables are N_1, N_2, R and $V_{B.}$ Especially the latter two design variables reflect the quantity of liquid and vapor returned to the column respectively (e.g., Douglas, 1988).

Assume that the values of the feed composition X_F and the flow rate F are fixed (e.g., F = 100mol/hour, X_F = 0.45), then an analysis model is developed by our collaborators at the Petroleum Institute in UAE, which has N_1, N_2, R and V_B as inputs and B, X_B, D and X_D as outputs, represented by a black-box as in Figure 4.6.

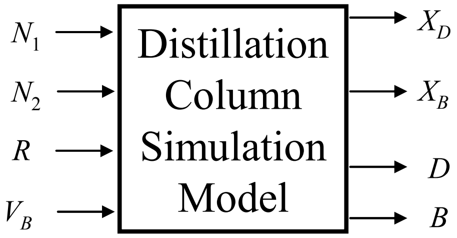
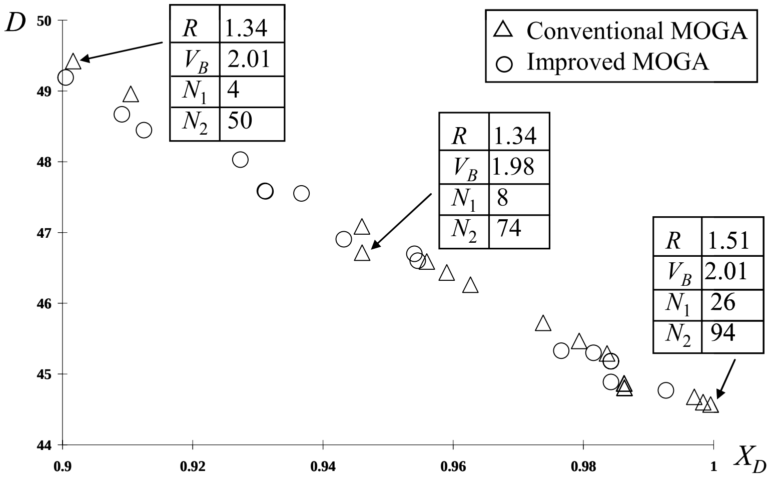


Figure 4.6 Inputs and outputs of the distillation column simulation model

The objectives in this problem are to simultaneously maximize the product yield (D) and purity (X_D), which are conflicting to each other. There are four design variables N_1, N_2, R, V_B. The lower and upper bound of the four design variables are given (see Eq. (4.3)). A multi-objective optimization model with two design objectives and several constraints is constructed:

$$\begin{aligned}
&\text{maximize } X_D = f_1(N_1, N_2, R, V_B)\\
&\text{maximize } D = f_2(N_1, N_2, R, V_B)\\
&\text{subject to:}\\
&0 < V_B \le 5\\
&0 < R \le 5\\
&1 \le N_1 \le 100\\
&1 \le N_2 \le 100\\
&0.9 \le X_D < 1\\
&0 \le X_B < 1\\
&B > 0\\
&D > 0
\end{aligned} \tag{4.3}$$

This optimization model is solved using the conventional MOGA and the improved MOGA. Figure 4.7 shows the obtained optimization results. It is observed that the Pareto frontiers from both approaches agree with each other well. In other words, it shows that the improved MOGA can be used to obtain optimum solutions for such problems with the same quality as that from the conventional MOGA. The results of the comparison in terms of the number of function calls for the improved MOGA, the conventional MOGA and the K-MOGA are given in Table 4.1 and Table 4.2.



R	1.34
V_B	2.01
N_1	4
N_2	50

R	1.34
V_B	1.98
N_1	8
N_2	74

R	1.51
V_B	2.01
N_1	26
N_2	94

Figure 4.7 Pareto solutions for distillation column design

4.5.2 Additional Test Examples

In this section, four additional examples, i.e., ZDT1, ZDT2, ZDT3, and Gear-Train from Deb (2001) are presented to further demonstrate the applicability of the proposed approach. These examples are also used in the last chapter and thus the formulations for these examples are already given in the last chapter. The Pareto frontiers obtained using the improved MOGA and the conventional MOGA are given in Figures 4.8-4.11.

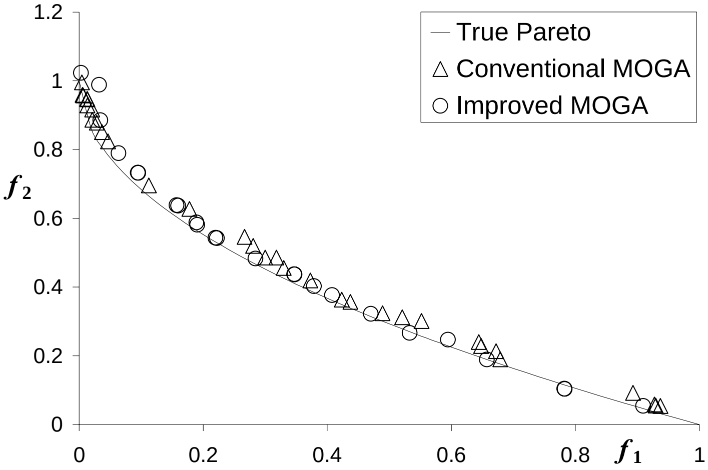


Figure 4.8 Pareto frontiers for ZDT1 example

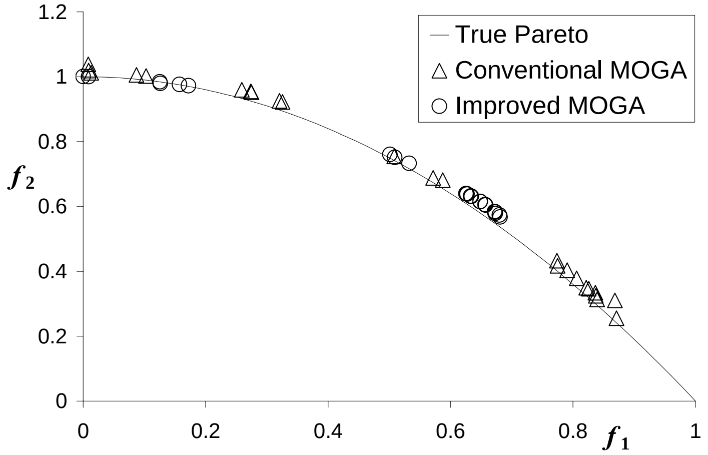


Figure 4.9 Pareto frontiers for ZDT2 example

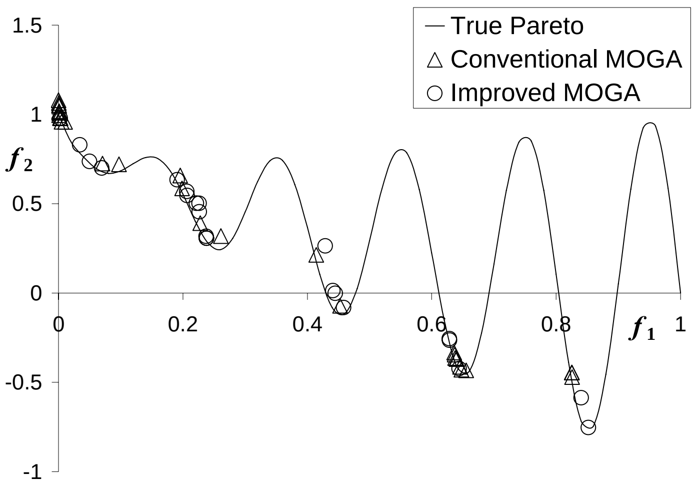


Figure 4.10 Pareto frontiers for ZDT3 example

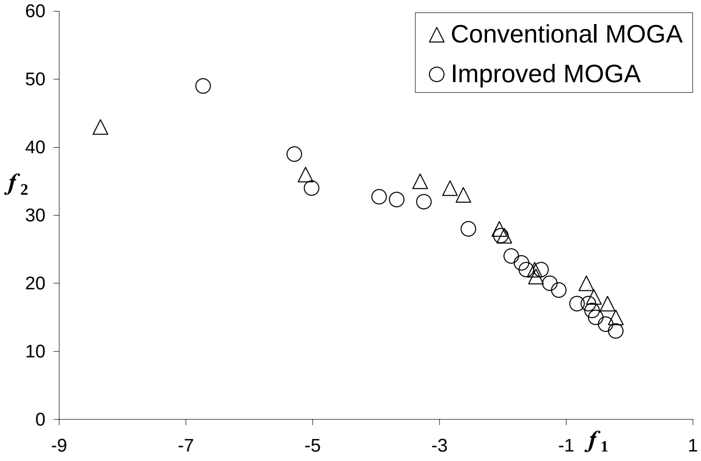


Figure 4.11 Pareto frontiers for Gear-Train example

It can be seen in Figures 4.8-4.11 that the Pareto frontiers obtained using the improved MOGA agree with those from the conventional MOGA for the four examples.

In terms of the number of function calls, the performance of the improved MOGA has been compared against the conventional MOGA and also the K-MOGA of Chapter 3. The results are presented in Table 4.1 where again (as in Chapter 3) each example was run 30 times to account for the stochastic nature of MOGA.

Table 4.1 Statistics for number of function calls

Example & Population size	Number of function calls								
	Conventional MOGA			K-MOGA			Improved MOGA		
	30 runs	Mean	STD	30 runs	Mean	STD	30 runs	Mean	STD
ZDT1 (30)	[293-594]	398	51	[120-258]	172	38	[73-129]	102	12
ZDT2 (30)	[231-561]	417	73	[134-197]	156	12	[55-103]	71	13
ZDT3 (30)	[340-616]	485	69	[176-307]	243	34	[121-176]	149	14
Gear Train (30)	[347-633]	471	78	[159-323]	234	39	[119-271]	203	45
Distillation Column (40)	[472-689]	599	54	[368-437]	396	37	[280-336]	315	31

Based on the data in Table 4.1, the savings obtained in terms of the number of function calls for each example are calculated based on their mean value. This calculation is performed for the K-MOGA over the conventional MOGA, the improved MOGA over the conventional MOGA, and the improved MOGA over the K-MOGA, separately. The results are shown in Table 4.2.

Table 4.2 Reduction of the average number of function calls

Example (Population size)	Reduction in terms in the mean value of the number of function calls		
	$1-\left(\frac{\text{K-MOGA}}{\text{Conventional MOGA}}\right)$	$1-\left(\frac{\text{Improved MOGA}}{\text{Conventional MOGA}}\right)$	$1-\left(\frac{\text{Improved MOGA}}{\text{K-MOGA}}\right)$
ZDT1 (30)	57%	74%	41%
ZDT2 (30)	63%	83%	54%
ZDT3 (30)	50%	69%	39%
Gear Train (30)	50%	57%	13%
Distillation Column (40)	34%	47%	20%
Average saving	47%	66%	33%

It is observed in Table 4.2 that the improved MOGA outperforms both the K-MOGA and the conventional MOGA in terms of the average number of function calls for each and all five examples. In particular, on the average, the proposed improved MOGA can save 66% in the number of function calls over the conventional MOGA, while the K-MOGA can save 47% over the conventional MOGA. In other words, the improved MOGA uses about 33% fewer number of the function calls over K-MOGA.

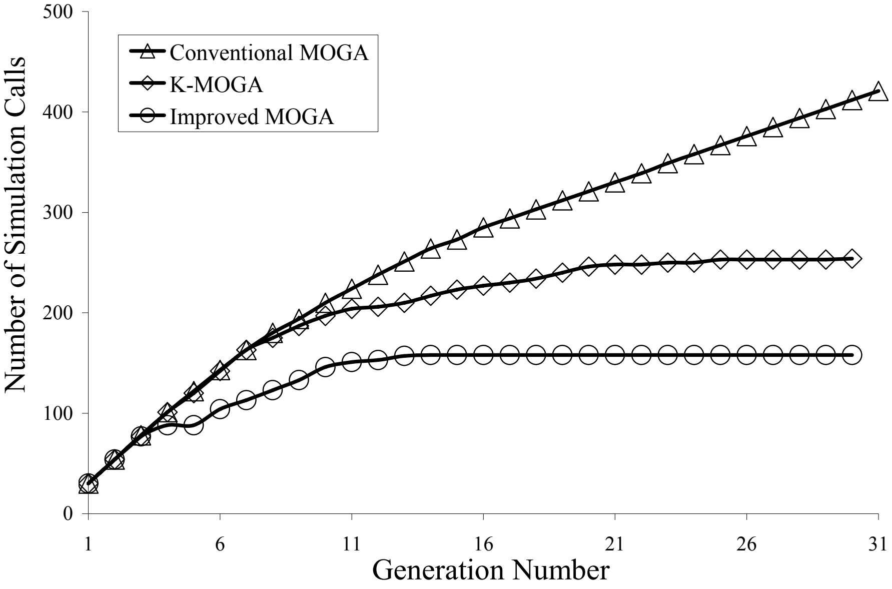


Figure 4.12 Number of generations for convergence

It can be seen in Figure 4.12 that, according to the stopping criterion introduced for the three approaches, the improved MOGA requires fewer generations for convergence than that for the K-MOGA. According to the first part of the stopping criterion in Section 3.3.4, it is observed that the conventional MOGA converges at about generation number 30 for this example. However, the curve for the conventional MOGA does not show flatness in Figure 4.12. As discussed in Section 3.3.4, this is because there are always new design points generated by the crossover and mutation operations in the conventional MOGA, and these design points can only be evaluated by the simulation. As a result, the number of function calls keeps increasing as the generation number increases, as shown in Figure 4.12. Similar results are obtained for the other examples.

4.6 Summary

The main contribution of the approach developed in this chapter is that for the first time in the literature online metamodeling for both fitness estimation and reproduction has been integrated and used to significantly reduce the number of function calls in a MOGA. As shown in the examples and test results, this improved MOGA has resulted in significant savings in terms of the number of function calls while obtaining solutions that are similar to those obtained from a conventional MOGA.

There are two main features in the improved MOGA. These are: i) online metamodeling for fitness evaluation, and ii) adaptive DOE for reproduction. These two features work in concert with the kriging metamodeling in different phases of the MOGA as follows.

In the fitness evaluation, kriging metamodeling is used to estimate the response for some design points to save the number of function calls. This is feasible because the responses for the design points do not have to be calculated precisely using the simulation. Instead, an estimate of their simulation response can be used only if the associated error does not change the domination status. In the reproduction phase, kriging metamodeling is used to predict the Pareto frontier in every generation and thus help MOGA to converge faster. As MOGA solutions are evolved, the predicted Pareto frontier approaches the Pareto frontier. Thus, when some design points are selected from the predicted optima and their responses are evaluated and added to the next population, the number of non-dominated design points is increased more quickly, so that eventually the Pareto frontier is obtained in fewer generations than with a conventional MOGA.

Five examples of both numerical and engineering types and with different degrees of difficulty have been used here to demonstrate the applicability of the improved

MOGA. The results confirm that this new optimization approach is indeed able to estimate the Pareto frontier with comparable quality to a conventional MOGA approach, while significantly reducing the number of function calls. For the five examples, it was observed that using the improved MOGA saved an average of 66% of function calls compared to the conventional MOGA.

The fitness estimation and adaptive DOE techniques proposed in this chapter are not restricted to a specific MOGA approach. In fact, these techniques can be integrated with other population-based stochastic multi-objective optimization methods with schemes for fitness evaluation and new population generation, such as the particle swarm optimization (e.g., Kennedy and Eberhart, 1995).

In the next chapter, the research thrust three, a novel adaptive DOE method in offline metamodeling, is presented.

Chapter 5 Adaptive DOE in Offline Metamodeling

5.1 Introduction

In chapters 3 and 4, the fitness estimation and DOE techniques were used in the context of online metamodeling in MOGA for design optimization. An alternative strategy to reduce the number of function calls is to use offline metamodeling (see, e.g., Barthelemy and Haftka, 1993; Myers and Montgomery, 1995; Roux et al., 1998; Ruzika and Wiecek, 2003; Simpson et al., 2004). The goal of the offline metamodeling is to construct a reasonably accurate metamodel using as few experiments as possible. Clearly, the metamodel accuracy is heavily dependent upon the DOE method used.

In this chapter, a new adaptive DOE approach is developed, which selects more points in non-smooth regions (see the definition in Section 5.3.1). We refer to the method as an *ACcumulative Error* (ACE) approach for DOE. In the ACE approach, the maximum entropy design method (recall Section 2.6) is used for initial design and kriging (recall Section 2.5) for metamodeling. In particular, the leave-one-out error (see Section 5.4.1) for existing experiments is obtained, the non-smooth regions (which contain local optima) are probed using this information, and the next point is sampled in one or more such regions. After that, the metamodel is updated and non-smooth regions are probed again. The procedure is repeated until potentially all non-smooth regions have some sampled points or no additional new non-smooth regions are discovered.

The rest of the chapter is organized as follows. A review of previous work for adaptive DOE methods is given in Section 5.2. In Section 5.3, terminology and definition used in the proposed approach are introduced. In Section 5.4, the new DOE approach is proposed and discussed in detail. In Section 5.5, several numerical and engineering examples and the corresponding results are discussed. Finally, the concluding remarks are given in Section 5.6.

A portion of this chapter was presented in Li and Azarm (2006).

5.2 Background for Adaptive DOE Methods

As introduced in Chapter 1, a DOE approach is said to be non-adaptive when the only information used in it is from the experiments. Examples of non-adaptive DOE approaches are: Latin hypercubes (McKay et al., 1979), maximum entropy design (Shewry and Wynn, 1987; Koehler and Owen, 1996), mean squared error (Jin et al., 2002), integrated mean square error (Sacks et al., 1989), maximin distance approach (Johnson et al., 1990), orthogonal arrays (Owen, 1992), and uniform design approaches (Fang et al., 1994, 2000). In general, a non-adaptive approach results in the same set of experiments whether or not it is applied in a single stage or sequentially in different stages of sampling.

On the other hand, a DOE approach is adaptive when the information from both the experiments and metamodel is used in selecting the next experiments. In general, sequentially adaptive approaches are considered to be superior to non-adaptive approaches in terms of improving the metamodel accuracy. A sequentially adaptive DOE approach often begins with an initial DOE that is generated using some non-adaptive or random approach. After that, the metamodel is constructed based on the initial experiments and the corresponding responses from the simulation. The information from the experiments and their responses together with the metamodel is used for sampling the next point or set of points. This procedure continues until a predefined stopping criterion is met.

Based on the adaptation scheme used, the sequentially adaptive DOE approaches can be classified into three types. The first type refers to the methods that are made adaptive according to the estimated optima (e.g., Watson and Barnes, 1995). The second type refers to the methods that are made adaptive based on the experiments that produce large uncertainty associated with an estimated response (e.g., SDO or Sequential Design for Optimization, see: Cox and John, 1992, 1997). The third type refers to the methods which are based on a compromise between the estimated optima and large uncertainty in the response (e.g., Schonlau et al., 1997; Jones et al., 1998; Sasena, 2002; Sasena et al., 2000; Farhang-Mehr and Azarm, 2002 and 2005; Farhang-Mehr et al., 2003).

All of the aforementioned approaches directly use an estimate for the simulation response and/or uncertainty in the estimated response in selecting new points. Unfortunately, such estimates are often inaccurate, especially when the number of experiments used to build the metamodel is small. Hence, these adaptive DOE approach may sample some experiments that are not desired: The selected points have neither a good estimate of the optima (maximum or minimum of an objective function) nor provide a good prediction of uncertainty in the estimated response. As a result, the sampling scheme in these methods is likely to consume precious function calls without significantly improve the metamodel accuracy.

To address the above mentioned shortcoming, a new DOE method is developed in this Chapter. This new DOE approach only uses true responses in determining non-smooth regions and selecting new points, thus it avoids being misled by the estimated uncertainty in the response during the early-stage of metamodeling.

5.3 Terminology and Definition

In this section, some terminology and definition used in the proposed adaptive DOE method are given.

5.3.1 Non-Smooth Region

A non-smooth region refers to a region in the input space where the responses for adjacent points usually have very different values. An example of a non-smooth region is shown in Figure 5.1, where $x \in (0.04, 0.33)$ corresponds to a non-smooth region, in which the corresponding response values (the curve inside the dashed circle in Figure 5.1) changes significantly. That is, a non-smooth region usually contains one or more local optima (maxima and/or minima). For the purpose of the proposed approach, we exclude the cases in which the response has an asymptotic behavior within the non-smooth region and that its value is infinite. Note that our definition for a non-smooth region is applicable to both continuous and discontinuous domains.

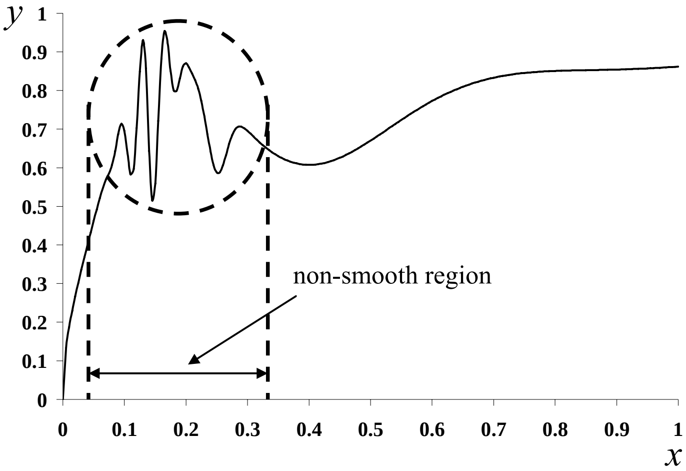


Figure 5.1 Non-smooth region

5.3.2 Greedy Algorithm

For completeness, this subsection provides a brief review of the principle of a greedy algorithm (e.g., Weiss, 1997), which is used later in the proposed adaptive DOE approach.

Greedy Algorithms are often used to quickly find global optimal solutions for a class of combinatorial optimization problems with the following two properties: 1) Greedy-choice property, which means the global optimal solution can be reached by solving a sequence of locally optimal choice sub-problems, and 2) Optimal substructure, which means an optimal solution to the sub-problem (locally optimal) is always a subset of the global optimal solution for the entire problem.

A classical example of an optimization problem with these two properties that can be solved quickly by the greedy algorithm is the problem of making a change (e.g., Cormen et al., 2001). Suppose you are asked to change a certain U.S dollar amount by using the smallest possible number of coins, where coins available are quarters (25 cents), dimes (10 cents), nickels (5 cents) and pennies (1 cent). As an example, suppose you buy a cup of hot chocolate that costs for 1.36 dollars. You give the cashier 2 dollars, thus the cashier owes you 64 cents. In order to give you back the smallest number of coins, the cashier calculates the change using a greedy algorithm as follows. First the cashier gives you two quarters, and then a dime, and finally he/she gives you four pennies. The total number of coins is seven, which is the smallest number of coins for 64 cents. This solution is a global optimal solution (i.e., the smallest number of coins needed). Note that this problem has the above-mentioned two properties. That is, every time the minimum number of coins with the largest value (local optimum) is chosen, and once a coin has been included in the solution set, it remains in the final solution set (globally optimal). In fact, it was proved (e.g., Weiss, 1997; Cormen et al., 2001) that a greedy algorithm can always yield the global optimum choice for such a class of problems.

In general, a greedy algorithm solves optimization problems in a sequence of steps: It works in a top-down fashion, making one greedy choice after another, reducing the optimization problem to a smaller one in each step. In the greedy algorithm, the procedure of choosing a local optimum is repeatedly performed with the hope that a desired global optimal solution is found at the termination, which contains all the explored local optima.

Using a greedy algorithm, an adaptive DOE problem can be posed in a way similar to the above problem of making a change: Select as few as possible experiments such that the metamodel accuracy can be improved as much as possible. Thus, an adaptive DOE problem can be solved using a greedy algorithm. That is, we first select every new experiment according to the current information from both existing experiments and the metamodel so that the metamodel accuracy is improved in the most inaccurate location. As a result, the final design (containing all selected experiments) is expected to improve the overall metamodel accuracy. Thus, the greedy algorithm is one way to tackle the adaptive DOE problem and can yield a very good, if not the best possible, design.

In the next section, a new adaptive DOE approach is proposed according to the greedy algorithm principle.

5.4 The ACcumulative Error (ACE) Approach for DOE

The proposed adaptive DOE approach is based on the observation that a reasonably accurate metamodel for simulation response can be constructed if all local optima of the response are evaluated. Moreover, the local optima are assumed to be located in the non-smooth regions which can be probed by calculating the leave-one-out error (e.g., Meckesheimer and Booker, 2002) by an experiment in such a region (see the next section).

In the following sections, the method of probing non-smooth regions using the leave-one-out error (LOO) is discussed first. Then, the ACcumulative Error (ACE) is introduced and its relation with the leave-one-out error is given. Finally, the proposed new DOE approach using the principles of the greedy algorithm and ACE is presented.

5.4.1 Leave-One-Out Error

Leave-One-Out (LOO) or cross-validation is often used for metamodel accuracy assessment as well as in sampling approaches (Jin et al., 2002). The basic procedure used in LOO is as follows. First, the method begins by leaving out any experiment from the current design ***D***. Next, the method fits a metamodel to the remaining experiments in ***D***. The method uses this metamodel to predict the response for the left-out experiment. Finally, the method calculates the difference between the predicted response (i.e., the metamodel response) and actual response (the simulation) for that left-out experiment. This procedure is performed for all experiments in ***D***, by leaving them out one at a time.

We define the LOO error for an experiment x_i as: $e_{LOO}^{x_i}$, which represents the difference between the actual response and its estimate:

$$e_{LOO}^{x_i} = \left| y(x_i) - \hat{y}(x_i) \right| \tag{5.1}$$

where $y(x_i)$ is the actual response for x_i, and $\hat{y}(x_i)$ is the corresponding metamodel response when x_i is left out. It is observed that the quantity $e_{LOO}^{x_i}$ can be used to represent the fidelity of the metamodel in a small region around x_i, as shown in Figure 5.2 and Figure 5.3.

In Figure 5.2, a metamodel with 10 experiments is constructed. Following the definition in Section 5.2, it is observed that the experiment: *x* = 0.125, is in a non-smooth region while *x* = 0.625, is not in a non-smooth region.

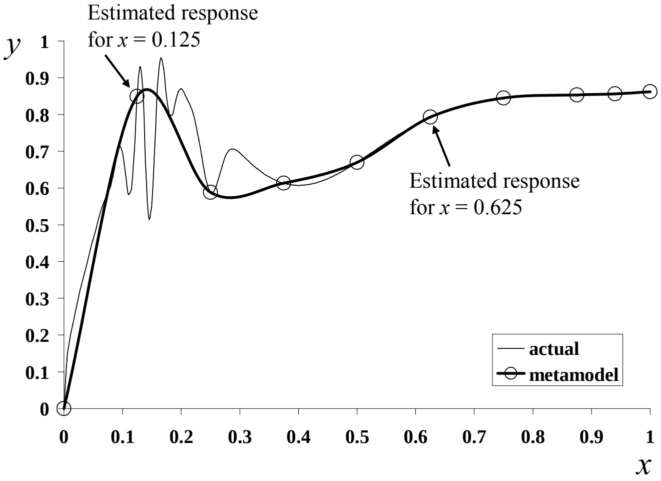
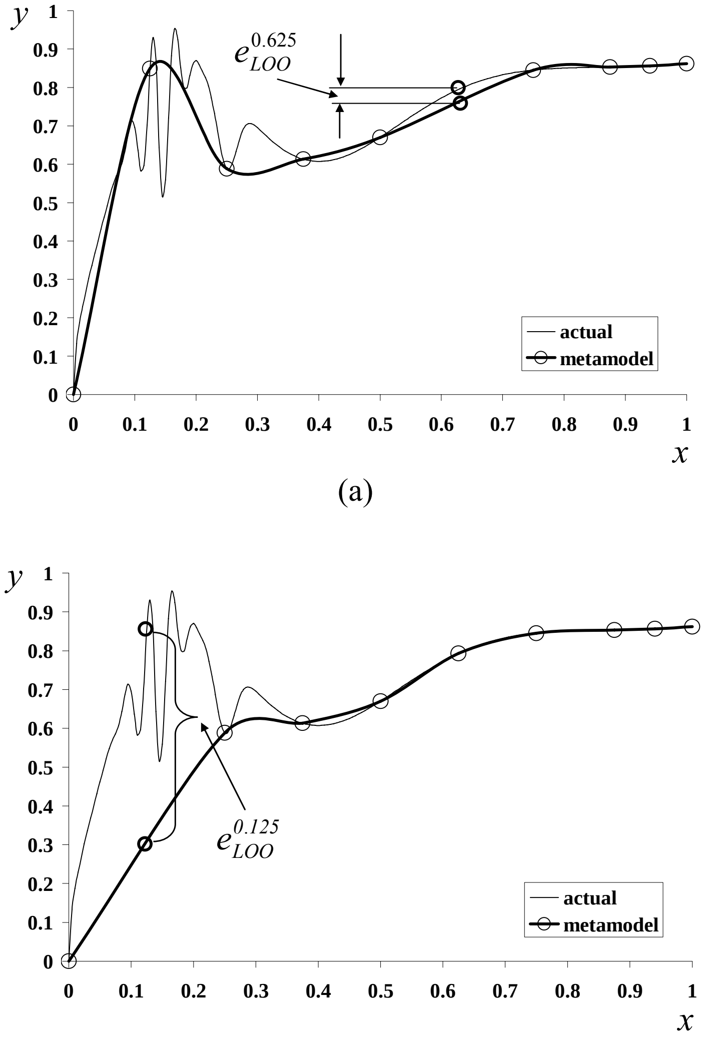


Figure 5.2 Metamodel with 10 experiments

Two metamodels, each with a left-out experiment, are built and shown in Figure 5.3(a) and Figure 5.3(b), separately. It can be seen in Figure 5.3(a) that a small value of $e_{LOO}^{x_i}$ (i.e., $e_{LOO}^{0.625}$) indicates that either the metamodel in the region around this point is smooth and has a good accuracy or the true response for this point is estimated well by the metamodel even if this point is in a non-smooth region. According to the definition of the LOO error, the latter situation implies that there are already enough experiments in the non-smooth region around this point and no additional experiments are needed.

The former implies that the region is already smooth and not many sample points are needed. On the other hand, as shown in Figure 5.3(b), a large value of $e_{LOO}^{x_i}$ (i.e., $e_{LOO}^{0.125}$) means that the response in this region is more likely to be non-smooth or not sampled well and thus more experiments are needed to improve the metamodel accuracy. Hence, the value of $e_{LOO}^{x_i}$ of an existing experiment can be used to identify regions where more points need to be selected to improve the metamodel accuracy.



(a)

(b)

Figure 5.3 Metamodels with point (a) *x* = 0.625 left out, and (b) *x* = 0.125 left out

Note that the value of $e_{LOO}^{x_i}$ is dependent upon the number or location of the remaining experiments (e.g., 9 experiments remained after one point left out) in the current design ***D***. This dependency is useful for updating the values of the LOO error when more experiments are available, thus making possible to identify some new non-smooth regions.

The goal of the DOE in this chapter is to sample new points that are not in the current design ***D***. However, the values of the LOO error are not available for those unobserved points. Thus, it can not be determined whether an unobserved point is in a non-smooth region can not be obtained directly by calculating the value of the LOO error.

In the next subsection, the ACE concept is introduced. The ACE concept will be used to estimate the value of the LOO error for an unobserved point, so that additional points can be sampled according to the estimated values of the LOO error.

5.4.2 Degree of Influence and Accumulative Error

The metamodeling approach used in this chapter is based on the kriging (see, e.g., Section 2.5). In the kriging metamodeling, the estimated response for an unobserved point is assumed to be highly correlated to the responses from the experiments that are close to that point. If an existing experiment is located in a non-smooth region in which only a few experiments have been sampled, then an unobserved point close to this experiment will likely have a large value of the LOO error. In this regard, the LOO error of an unobserved point x_0 can be estimated by the values of the LOO error for all existing experiments around this unobserved point. Note that the experiments that are close to x_0 should have more of an impact in terms of the LOO error on x_0 than those which are relatively far from it.

We define the Degree-of-Influence (*DoI*) from x_i to x_0 as a function of the distance form x_i to x_0. In essence, *DoI* represents the amount of influence that the LOO error has from x_i on x_0. Many different functions are introduced in the literature to represent this kind of correlation (e.g., Lophaven, 2002; Clark, 2005). In this chapter, we use an exponential function:

$$DoI(x_i) = \exp\left(-\alpha \left\| x_i - x_0 \right\|\right) \tag{5.2}$$

where α determines how fast *DoI* (x_i) diminishes as x_i moves away from x_0. The value of *DoI* (x_i) diminishes rapidly even for a small distance when the value of α is large. Using Eq. (5.2), the relationship between *DoI* and $\left\| x_i - x_0 \right\|$ with different values of α is shown in Figure 5.4.

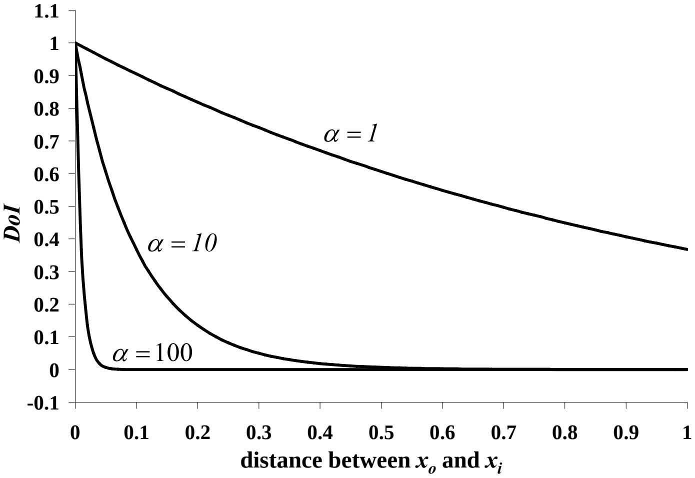


Figure 5.4 *DoI* vs. $\left\| x_i - x_0 \right\|$ for different α values

It is observed in Figure 5.4 that for $\alpha = 100$, the value of *DoI* is almost zero when the distance between x_i and x_o is greater than about 0.08. On the other hand, for

$\alpha = 10$, the distance can be as large as about 0.7 for the value of *DoI* to be almost zero, where "almost zero" means less than 10^{-5}, which is considered as the worst case *DoI* here. Any value of *DoI* which is less than this worst case *DoI* is discarded.

The value of α actually defines a hyper-sphere, i.e., a *DoI* region, where an experiment x_i resides in the center of the region and has a non-zero *DoI* (i.e., greater than or equal to the worst case *DoI*) for an unobserved point x_0 located in this region (but not in its center). In other words, the value of α should be determined such that any unobserved point x_0 will be placed in at least one *DoI* region. Thus every x_0 will have to have a non-zero *DoI*. The value of α can be obtained using a worst case analysis: When x_0 is on the boundary of the *DoI* region of an experiment that is closest to x_0, the value of *DoI* from this experiment should be non-zero (i.e., greater than or equal to the worst case *DoI*). In this way, the radius of the hyper-sphere is defined as the *DoI* distance, and this value will be used for all experiments in the current design ***D***. The *DoI* distance is obtained using a three-step max-min procedure[1]: For a given design ***D*** with *n* experiments, Step 1: For every experiment from design ***D***, calculate the minimum distance from that experiment to all other experiments in ***D***. This will obtain *n* number of minimum distances. (Note that the number may be less than *n* since some of the minimum distances might be the same.) Step 2: Obtain the maximum distance out of the *n* minimum distances. Step 3: Set *DoI* distance to be one half of the maximum distance, i.e., the radius of the hyper-sphere. Note that these steps are performed based on the normalized values of x_i and x_0. The lower and upper bounds

[1] Note that the computational complexity of this procedure is: $O(n^2)$, where *n* is the number of experiments in the design space.

in the input space can be used to normalize the input points to [0, 1].

Having the values of the worst case *DoI* (e.g., 10^{-5}) and *DoI* distance (see Step 3 in the last paragraph), based on Eq. (5.2), the value of α can be obtained:

$$\alpha = \frac{-\ln\left(\text{worst case } DoI\right)}{DoI \text{ distance}} \tag{5.3}$$

Note that the value of α may not change in every iteration since a newly selected experiment may not change the *DoI* distance.

Being able to obtain α, the estimated LOO error for an unobserved point x_0 is calculated next:

$$e_{AC}^{x_0} = \sum_{i=1}^{n} e_{LOO}^{x_i}\left[DoI\left(x_i\right)\right] \tag{5.4}$$

where $e_{LOO}^{x_i}$ is obtained in Eq. (5.2), $e_{AC}^{x_0}$ gives the ACE value for x_0, and x_i is an existing experiment. In essence, Eq. (5.4) implies that the ACE value for an unobserved point is determined based on the LOO error and the proximity of experiments around this point. Substituting Eq. (5.2) into Eq. (5.4) results in Eq. (5.5):

$$e_{AC}^{x_0} = \sum_{i=1}^{n} e_{LOO}^{x_i}\left[\exp\left(-\alpha\left\|x_i - x_0\right\|\right)\right] \tag{5.5}$$

It can be seen in Eq. (5.5) that an unobserved point can have a large value of $e_{AC}^{x_0}$ if it is in a region where existing experiments have a large value of the LOO error. Such a region is indicated as a non-smooth region and potentially does not have enough sampled experiments, and thus is a good candidate region for sampling new points.

5.4.3 A Greedy Algorithm for Adaptive DOE

In this subsection, we present the proposed ACE approach with several parts.

These are: 1) Criterion for selecting new points, 2) initial design and stopping criterion, 3) steps for the proposed approach, and 4) computational complexity of the approach.

1) Criterion for Selecting New Points

Now that the estimate of the LOO error (and ACE) for an unobserved point can be calculated, given i experiments, the new (i+1)th experiment, or x_{i+1}, is obtained by selecting a point with the largest value of the ACE, i.e.,

$$x_{i+1} = \text{arg max } e_{AC}^{x_0},\ x_0 \in U,\ x_0 \notin \boldsymbol{D}, x \in \boldsymbol{D} \tag{5.6}$$

where "arg max" denotes the optimal solution for the optimization problem in which the value of $e_{AC}^{x_0}$ is maximized.

It should be clear that the point x_{i+1} obtained using Eq. (5.6) tends to be very close to some existing experiments in ***D***. This is because of the way $e_{AC}^{x_0}$ is calculated: The closer x_0 is to some existing experiments, the larger the value of $e_{AC}^{x_0}$. As a result, the newly selected points tend to cluster in a non-smooth region first, and then another and so on. However, note that the number of available experiments is limited in real engineering problems and thus some non-smooth regions may not even get the chance to be sampled. Hence, instead of heavily sampling one non-smooth region first, we will try to sample all non-smooth regions with the same degree of importance. To do this, the concept of a *cluster threshold* is introduced. This threshold is used to prevent clustering of new points with existing experiments.

A cluster threshold is obtained in a three-step procedure which is somewhat similar to how the *DoI* distance is obtained. These steps are as follows. Step 1: For every experiment in the design ***D***, calculate the minimum distance from that experiment to all other experiments in ***D***. Step 2: Obtain the average for the obtained minimum distances. Step 3: Set the value of the cluster threshold to be one half of this average distance. The “one half” coefficient indicates that the new point can be located among experiments that have a greater distance to each other than the average distance obtained in Step 2. Note that the value of the cluster threshold changes when a new experiment is sampled.

For regions that have more experiments (i.e., clustered), the distance between the new point and each experiment is expected to be smaller than the average distance (see Step 3 in the last paragraph). Such a new point can not be sampled among these experiments since it violates the cluster threshold.

Considering the cluster threshold and substituting Eq. (5.5) into Eq. (5.6), a constrained maximization problem is obtained as follows:

$$\begin{aligned} & x_{i+1} = \arg\max \sum_{i=1}^{n} e_{LOO}^{x_i} \left[\exp\left(-\alpha \left\| x_i - x_0 \right\| \right) \right] \\ & \text{subject to}: \\ & \left\| x_i - x_0 \right\| \geq \textit{cluster threshold}, x_0 \in U, x_0 \notin \boldsymbol{D}, x_i \in \boldsymbol{D} \end{aligned} \tag{5.7}$$

where the inequality constraint is used so that the newly sampled point has a reasonable distance from all existing experiments.

As new points are selected and evaluated, and the kriging metamodel is updated, the values of ACE in the current non-smooth regions are reduced because the LOO error becomes smaller. On the other hand, these new points will also help explore new non-smooth regions according to the updated metamodel. Accordingly, the ACE approach will tend to sample points in the new non-smooth regions, since the values of ACE in such regions are expected to be large.

Even if a cluster threshold is used in the proposed approach, it is still possible that too many experiments are selected in some non-smooth regions, while other non-smooth regions are left un-sampled. This happens because: 1) Additional local optima are discovered in the current non-smooth regions as more points are sampled in these regions (an example will be discussed in Section 5.5.1) even if new non-smooth regions are discovered, and 2) no new non-smooth region has been explored. For both cases, the newly selected point is close to other existing experiments and the corresponding $e_{LOO}^{x_i}$ value can become very small (i.e., the sampling of the new point does not provide much improvement in the metamodel accuracy). In this regard, the maximum entropy design method (recall Section 2.6) is used in the proposed approach to sample the next new point and discover a potentially new non-smooth region. As mentioned before, the maximum entropy design method is a non-adaptive DOE approach since no information from the response is used for sampling subsequent experiments. Moreover, the maximum entropy design approach emphasizes selecting points in regions where fewer experiments have been sampled.

2) Initial Design and Stopping Criterion

By combining the greedy sampling criterion with an adaptively obtained cluster threshold as well as the maximum entropy design method (particularly when experiments are clustered), the proposed approach is able to sample multiple non-smooth regions, even if there are several of them, in the input space. However, such a capability is dependent upon an a priori discovery of a non-smooth region or regions.

A non-smooth region can be discovered in the initial design or in the process of sampling new points: For each experiment, a large value of the LOO error indicates that the corresponding experiment is potentially in a non-smooth region. In this regard, a basic requirement for an initial design is that the entire input space should be sampled uniformly so that all regions have an equal chance to be probed for the initial design. As discussed above, the maximum entropy design method is appropriate for an initial design since it tends to cover the entire input space symmetrically. Also note that the number of points in an initial design is dependent on the dimension of the input space: The higher the dimension of the input space, the more the number of points is required for the initial design.

So far, the proposed DOE approach is open-ended: It can sample as many experiments as needed, given that there are enough computational resources. However, in reality, the computation resources are limited. Thus, a stopping criterion should be included to terminate the sampling procedure. A DOE approach often stops according to: 1) The required metamodel accuracy (e.g., RMSE and MAE; or using cross-validation), or 2) the number of available simulation calls known a priori from the limitation in computational resources (with the hope that the obtained metamodel meets the requirement for accuracy). The metamodel accuracy is not used as a stopping criterion here simply to avoid extra simulation runs; rather, the number of available simulation calls is used as the stopping criterion.

3) Steps for the Proposed Approach

Figure 5.5 gives the flowchart for the proposed approach. The detailed steps are given in the next paragraph, which are grouped into initial steps and main steps.

Initial Steps:

Step 1: Initial design. In this step, *n* experiments are selected using the maximum entropy design method. The quantity *n* is used to account for the number of experiments in the design ***D***. Initially, *n* is set to be the number of experiments in the initial design.

Step 2: Get response values for the initial design.

Step 3: Build kriging metamodel for the initial design. The initial kriging metamodel is built based on the response values for the initial design. This metamodel is denoted the initial metamodel.

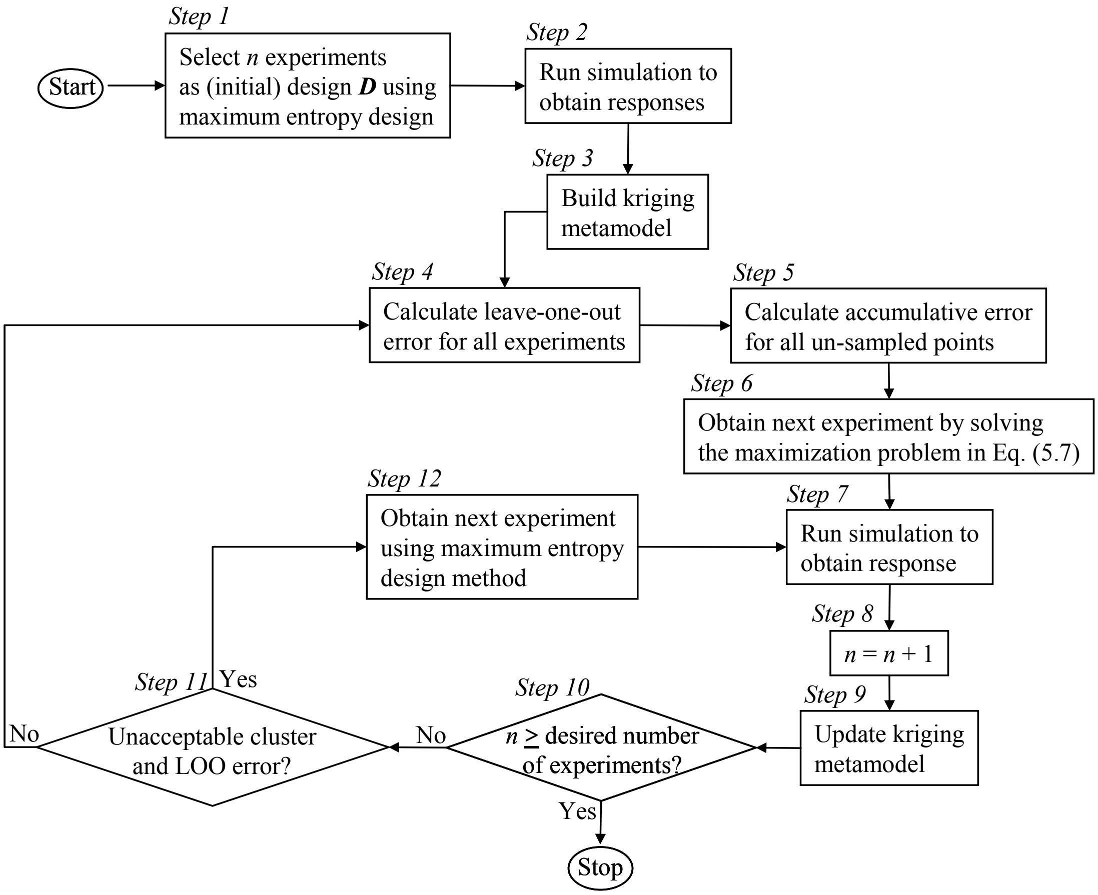

Start
Step 1
Select n experiments as (initial) design D using maximum entropy design
Step 2
Run simulation to obtain responses
Step 3
Build kriging metamodel
Step 4
Calculate leave-one-out error for all experiments
Step 5
Calculate accumulative error for all un-sampled points
Step 6
Obtain next experiment by solving the maximization problem in Eq. (5.7)
Step 12
Obtain next experiment using maximum entropy design method
Step 7
Run simulation to obtain response
Step 8
n = n + 1
Step 9
Update kriging metamodel
Step 10
n ≥ desired number of experiments?
Yes
Stop
No
Step 11
Unacceptable cluster and LOO error?
Yes
No

Figure 5.5 Flowchart for the proposed approach

Main Steps:

Step 4: LOO error calculation for the *n*th metamodel. The value of the LOO error for each experiment in the current design is obtained using Eq. (5.1).

Step 5: Accumulative error calculation for the *n*th metamodel. The value of the accumulative error for all un-sampled points are obtained using Eq. (5.5) based on the values of the LOO error from Step 4. (Note that the un-sampled points refer to all points in the normalized input space except the experiments.)

Step 6: Select the next experiment. The Eq. (5.7) is solved to obtain the next experiment.

Step 7: Obtain the response for the new experiment. The response for the newly selected experiment is obtained by the simulation.

Step 8: Increase the counter for the number of experiments by one. Set: $n = n + 1$.

Step 9: Update the kriging metamodel. The kriging metamodel is updated using the response for the nth experiment (i.e., the previous n experiments plus the new experiment).

Step 10: Check stopping criterion. Check whether n reaches the limitation of the number of experiments: If yes, stop the algorithm; otherwise, continue.

Step 11: Check the cluster and LOO error condition. When the nth experiment is very close to other experiments (i.e., closer than a presumed cluster threshold) and has a small LOO error, it is unacceptable and continue to the next step; otherwise, go to Step 4.

Step 12: Get the next experiment using the maximum entropy design method, and go back to Step 7.

As discussed in Section 5.3.2, an adaptive DOE problem can be proposed as an optimization problem which has the greedy-choice property and optimal substructure. Hence, a greedy algorithm can be used to find global optimal solution for such problems. On the other hand, the proposed approach follows the greedy principle because at every stage it chooses the new point with the largest value of ACE, which is obtained only based on current information, while the consequences for choosing this point is not considered.

In the next subsection, an analysis of the computational complexity is performed for the proposed approach.

4) Computational Complexity Analysis

Assume that the computational time for running a single simulation call is t_s, for kriging metamodeling is t_k, and for other operations (i.e., the leave-one-out error calculation) is t_o, with the assumption that $t_s >> t_k > t_o$. Also we define *n* as the size of current design, and *N* as the total number of all possible points in an input grid space. Following the procedure for the complexity analysis (e.g., Kozen, 1992; Cormen, 2001) and using the '*O*' notation (i.e., worst case analysis in terms of "order of magnitude" of calculations), a computational complexity analysis for the proposed approach can be performed as follows. In the step 4 of last section, the leave-one-out error for all experiments based on k^n (i.e., the kriging metamodel constructed with *n*th experiments) are calculated using Eq. (5.1), and the computational complexity is $O(n \times t_o)$. In the step 5, the accumulate error $e_{AC}^{x_0}$ for all unobserved points are obtained using Eq. (5.5), and the computational complexity is $O(N \times t_k)$. In the step 6, the next experiment is selected by solving the maximization problem in Eq. (5.7), and the computational complexity is $O(N \times t_o)$. In the step 7, the simulation is called to obtain the actual response, and the computational complexity for all simulation calls is $O(n \times t_s)$. Hence, the total computational complexity for the proposed approach is

$O(n \times t_o)$ + $O(N \times t_k)$ + $O(N \times t_o)$ + $O(n \times t_s)$. The total computational complexity is dominated by $O(n \times t_s)$ since it is assumed that $t_s >> t_k > t_o$, In other words, the total time for running the simulation contributes the most to the total computational cost of the approach.

Note that the computational cost of the proposed approach is independent of the dimension of the input space, in the sense that only the Euclidian distance of experiments is used to obtain the next experiment as shown in Eq. (5.7). In other words, a higher dimension input space only causes some more internal computations (e.g., t_k and t_o), but does not change the way of solving the maximization problem of Eq. (5.7).

5.5 Examples and Results

Most of the test problems used in this section have one or two inputs and one output so that the response surface/curve can be shown graphically.

First we use a numerical example with a single input to demonstrate the proposed approach during different sampling stages. After that an engineering example with two inputs is used to show the applicability of this new DOE approach to multi-dimensional engineering problems. Then, a two-dimensional-input numerical example with multiple non-smooth regions is used to demonstrate that the proposed approach is able to sample multiple non-smooth regions. Finally an engineering example with four design variables is used to show that the proposed approach is applicable to problems with multiple inputs. In all four examples, the kriging metamodeling (see, e.g., Section 2.5) is used for metamodel construction. The maximum entropy design approach (see, e.g., Section 2.6), which is a non-adaptive DOE approach, and the Sequential Design for Optimization (SDO) (Cox and John, 1992, 1997), an adaptive DOE approach, are used and compared with the proposed DOE approach in terms of the metamodel accuracy.

In the SDO approach, in order to select new experiments, the function $\hat{y}(x_0) - ks(x_0)$ is constructed and minimized, where x_0 is an unobserved point, $\hat{y}(x_0)$ is the estimated response for x_0, $s(x_0)$ is the standard deviation for the variance of the estimated response for x_0, and *k* is a scalar greater than zero. Each x_0 corresponds to a solution for the above function and the x_0 which gives the global minimum solution is selected as the next experiment. Following a previous work suggestion (Jones, 2001), the value of *k* is set to be 5 here. The SDO approach emphasizes on searching relatively un-sampled regions in the input space where the variance is large. See Cox and John (1992, 1997) for a more detailed description for the SDO approach.

5.5.1 Numerical Example 1: Single Input-Single Output, Multi-Modal

The numerical example is from Farhang-Mehr and Azarm (2002). It is a single-input (*x*) single-response (*y*) function as formulated in Eq. (5.8).

$$\begin{aligned} y &= (1 - e^{-2\sqrt{x}}) + 6xe^{-7x}\sin(10x) - 0.2e^{-2000(x-0.25)^2} \\ &\quad + 60\min(0, |x - 0.14| - 0.08)^2[\ln(x + 0.2) + 1.5\sin^2(85x)] \end{aligned} \tag{5.8}$$

We begin with an initial design of 8 experiments sampled using the maximum entropy design: *D* = {0, 0.125, 0.25, 0.375, 0.5, 0.625, 0.75, 1}, which includes experiments on the boundary and middle regions of the input space. The metamodel with 8 initial experiments (hollow points) and the actual response curve are shown in Figure 5.6.

It can be seen in Figure 5.6 that there is a non-smooth region for the response curve, which makes this function an interesting test problem for the proposed DOE approach. For this example, as discussed next, 27 additional experiments are used to build the metamodel, one by one. It is assumed that in this example the total number of available function calls is 35.

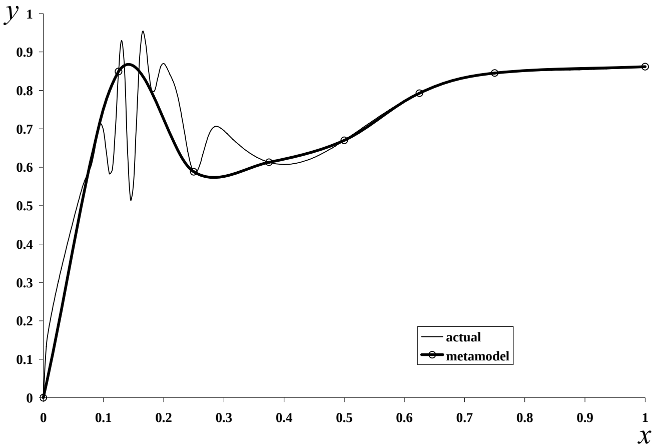
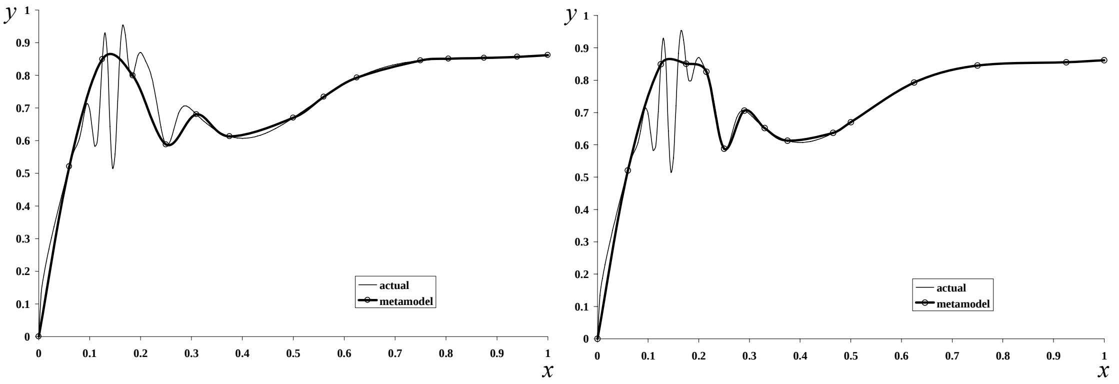


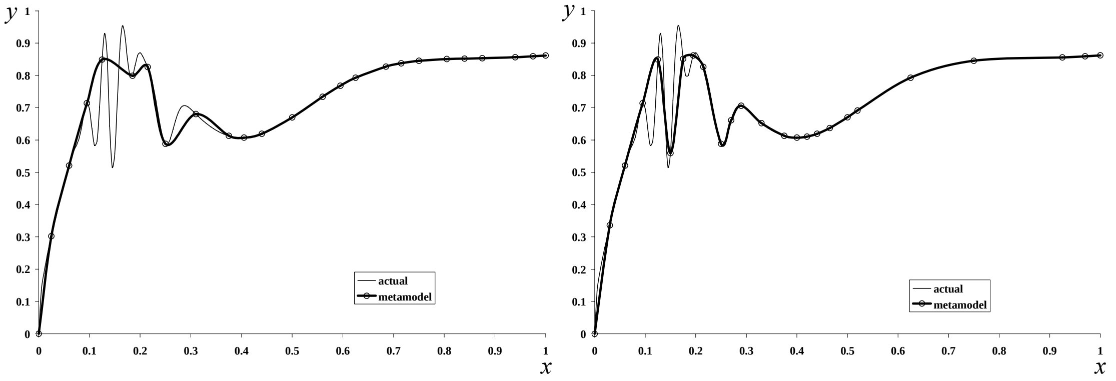
Figure 5.6 Metamodel using initial design

The experiments obtained using the new approach and the maximum entropy design approach. The corresponding metamodels are shown in Figure 5.7, Figure 5.8 and Figure 5.9, for 15 experiments, 25 experiments and 35 experiments, separately. And the values of RMSE and MAE (obtained using 1000 randomly selected testing points) are compared for the metamodels with 35 experiments using both approaches. It is observed that the proposed approach provides a more accurate metamodel in which both RMSE and MAE are improved. The metamodels with 15 and 25 experiments are used to show the progress in the metamodel accuracy of the approach as more experiments are selected.



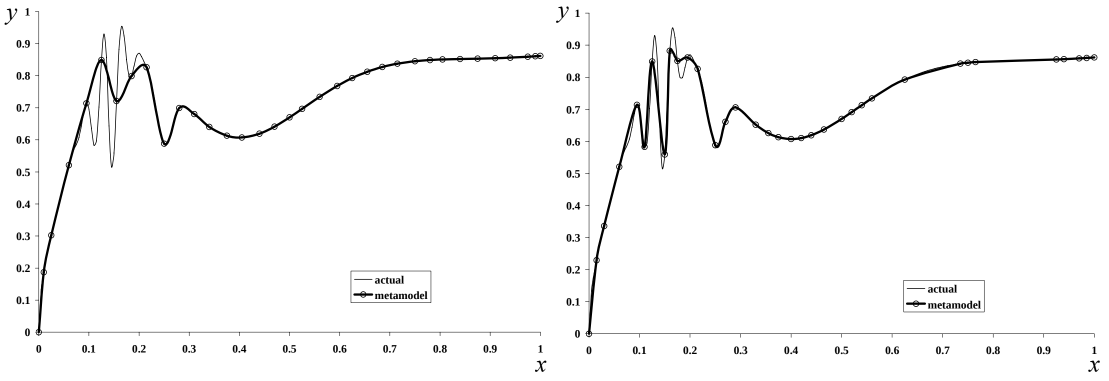
(a) (b)

Figure 5.7 Metamodels with 15 experiments for (a) maximum entropy design and (b) proposed approach



(a) (b)

Figure 5.8 Metamodels with 25 experiments for (a) maximum entropy design and (b) proposed approach



(a) (b)

Figure 5.9 Metamodels with 35 experiments for (a) maximum entropy design approach: RMSE = 0.067, MAE = 0.242 and (b) proposed approach: RMSE = 0.048, MAE = 0.115

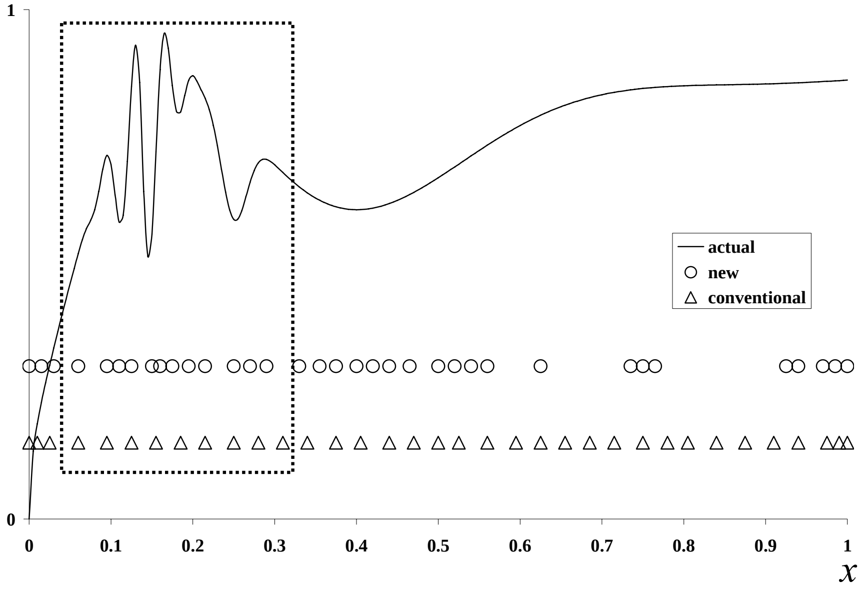


Figure 5.10 Final design for maximum entropy design and proposed approach

The 35 experiment designs sampled using the maximum entropy design and the proposed approach are shown in Figure 5.10. It can be observed that the proposed approach samples more points (12 of them) in the non-smooth regions than the maximum entropy design approach (9 of them). This observation confirms that the proposed approach is adaptive to the responses and hence it can help to construct a metamodel with high accuracy. This one-dimensional numerical example illustrates the applicability of the proposed DOE approach. It can be seen in Figure 5.7, Figure 5.8 and Figure 5.9 that the proposed approach tends to sample more points in the non-smooth regions, while the maximum entropy design approach keeps sampling the entire input space equally.

The experiments obtained using the maximum entropy design and the proposed approaches are shown in Table 5.1. The corresponding metamodel accuracy (RMSE and MAE) is calculated for the initial design and for every 5 additional experiments.

Table 5.1 Experiments and the corresponding metamodel accuracy

Experiment Number	Maximum Entropy Design Approach			Proposed Approach		
	Experiments	RMSE	MAE	Experiments	RMSE	MAE
8 Initial Experiments	0	0.272	0.322	0	0.272	0.322
	0.25			0.25		
	0.5			0.5		
	0.75			0.75		
	1			1		
	0.125			0.125		
	0.625			0.625		
	0.375			0.375		
9	0.875	0.243	0.322	0.925	0.237	0.322
10	0.94			0.06		
11	0.06	0.152	0.322	0.175	0.122	0.322
12	0.56			0.33		
13	0.31			0.29		
14	0.805			0.215		
15	0.185			0.465		
16	0.44	0.128	0.322	0.03	0.095	0.268
17	0.685			0.095		
18	0.975			0.97		
19	0.025			0.15		
20	0.405			0.4		
21	0.84	0.103	0.319	0.44	0.062	0.268
22	0.215			0.195		
23	0.595			0.27		
24	0.095			0.42		
25	0.715			0.52		
26	0.91	0.088	0.319	0.355	0.050	0.127
27	0.34			0.54		
28	0.525			0.56		
29	0.01			0.015		
30	0.99			0.11		
31	0.155	0.067	0.242	0.94	0.048	0.115
32	0.655			0.985		
33	0.28			0.735		
34	0.78			0.765		
35	0.47			0.16		

It is observed from the results shown in Table 5.1 that the proposed approach outperforms the maximum entropy design approach every time in terms of RMSE after the initial design. For MAE, the proposed approach also outperforms the maximum entropy design approach.

In order to compare the proposed approach with the SDO approach, the same 8 experiments: $D = \{0, 0.125, 0.25, 0.375, 0.5, 0.625, 0.75, 1\}$, are also used as an initial design. The SDO approach is used to obtain the additional 27 experiments one-by-one. The final design and metamodel are shown as below.

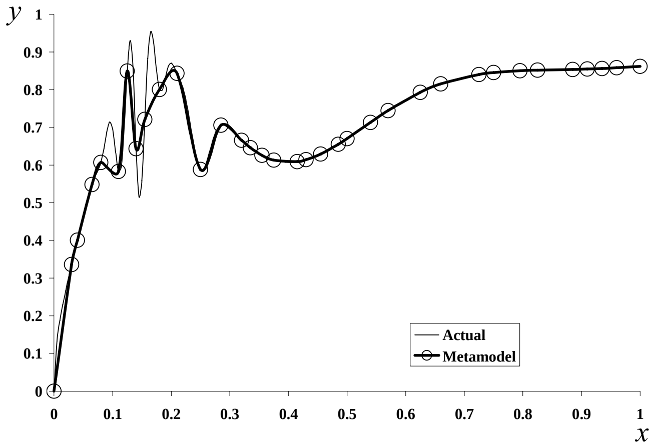


Figure 5.11 Design of 35 experiments and metamodel using SDO with RMSE = 0.051, MAE = 0.197

Comparing Figure 5.9 and Figure 5.11, it can be observed that: (i) the proposed approach and the SDO approach show similar values of RSME, while the proposed approach gives a smaller MAE, and (ii) both the proposed approach and the SDO approach show smaller values of RMSE and MAE than the maximum entropy design approach.

In order to study the effect of the number of initial experiments, we also applied the proposed approach on this example with 5 initial experiments sampled using the maximum entropy design: D = {0, 0.25, 0.5, 0.75, 1}, which includes experiments on boundary and middle regions of the input space. The metamodel obtained using the initial design is shown together with the actual response curve in Figure 5.12. It is observed that the metamodel in Figure 5.12 is different from that in Figure 5.6.

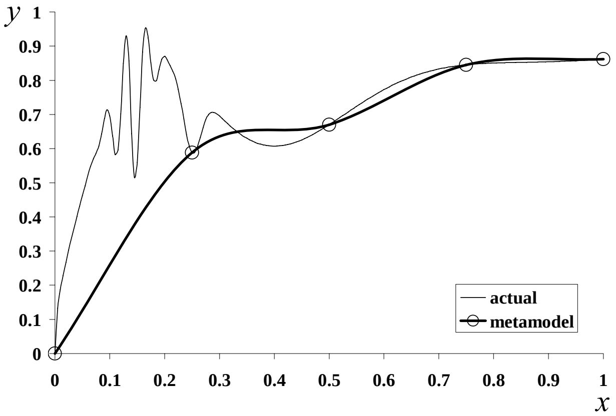


Figure 5.12 Metamodel using 5 initial experiments

The metamodel for 35 experiments are built and shown in Figure 5.13. A testing sample of 1000 randomly selected points is used to assess the metamodel accuracy by calculating RMSE and MAE.

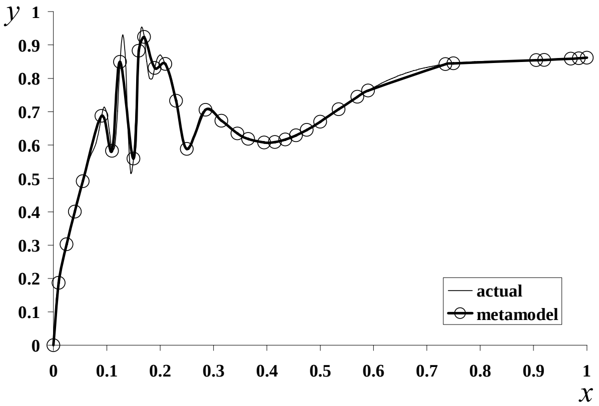


Figure 5.13 Metamodel for 35 experiments using 5 initial points

Comparing the metamodel in Figure 5.13 with that in Figure 5.9(b), it can be seen that they are similar to each other over the input space (*x*-axis), except that there are some minor differences. In fact, the values for both RMSE and MAE are quite similar to each other, i.e., RMSE = 0.045 and MAE = 0.115 for the metamodel with 5 initial experiments vs. RMSE = 0.048 and MAE = 0.115 for metamodel with 8 initial experiments. Thus, for this example, the number of initial experiments does not affect the accuracy of the final metamodel very much. However, this conclusion is tempered by the fact that in this example the number of initial experiments is much less than the number of experiments selected adaptively.

5.5.2 Numerical Example Two: Two-Input, Single Output, Multi-Modal

The second numerical example is from Jin et al. (2002). This numerical example has two inputs (i.e., x_1 and x_2), and one output (i.e., y). The function is given in Eq. (5.9) and the inputs are normalized to be in the range: [0, 1], and the actual response surface for the problem is shown in Figure 5.14 as follows:

$$y = 2 + 0.01\left(x_2 - x_1^{\,2}\right)^2 + \left(1 - x_1\right)^2 + 2\left(2 - x_2\right)^2 + 7\sin\left(0.5x_1\right)\sin\left(0.7x_1x_2\right) \qquad (5.9)$$

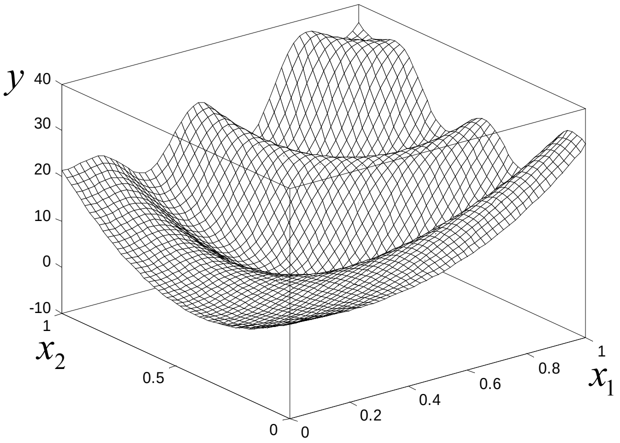


Figure 5.14 Actual response surface for the second numerical problem

We begin with an initial design with 25 experiments. Then we apply the proposed approach to obtain 25 additional experiments. The final design is shown in Figure 5.15(b). For comparison, the same procedure is performed using the maximum entropy design approach with a design that is in Figure 5.15(a).

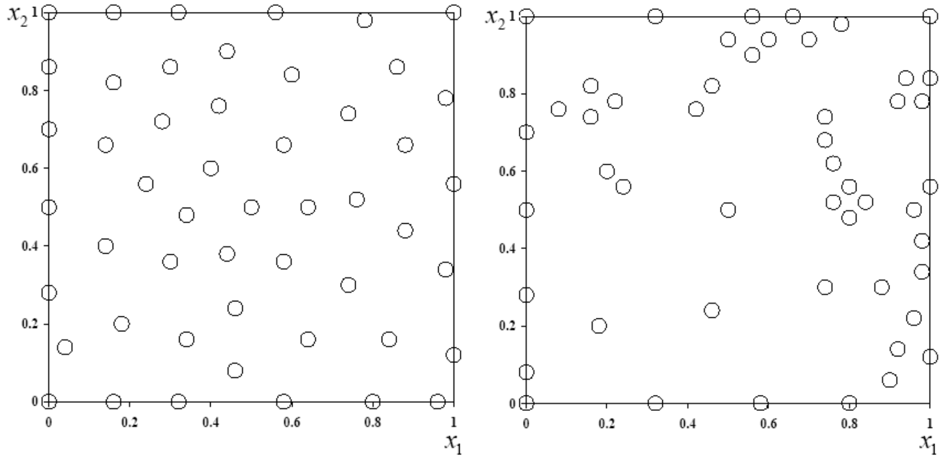
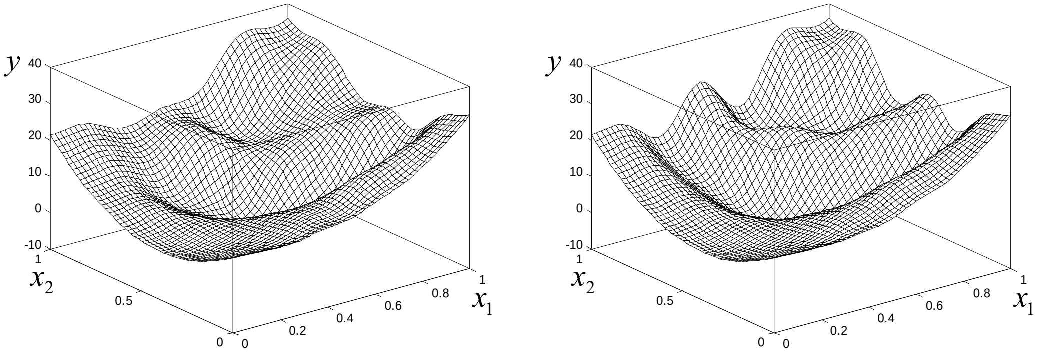


Figure 5.15 Final design: (a) maximum entropy design and (b) proposed approach

As observed in Figure 5.15, the proposed approach samples more points in the non-smooth regions than those by the maximum entropy design approach. This observation shows that the proposed approach is adaptive to the responses and hence can obtain a reasonably accurate metamodel.

The metamodels built based on the final designs are shown in Figure 5.16(a) and Figure 5.16(b).



(a)

(b)

Figure 5.16 Metamodels for (a) maximum entropy design approach with RMSE = 0.046, MAE = 0.268, and (b) proposed approach with RMSE = 0.020, MAE = 0.108

An additional testing sample of 1000 randomly selected experiments is used to assess the overall metamodel accuracy by calculating RMSE and MAE. Comparing the metamodels in Figure 5.16(a) and Figure 5.16(b), it can be observed that the proposed approach provides a more accurate metamodel in which both RMSE (normalized) and MAE (normalized) are improved compared to the maximum entropy design approach. Note that a metamodel with reasonable accuracy is important in the offline metamodeling assisted optimization of deterministic simulation models (e.g., Farhang-Mehr and Azarm, 2005). This numerical problem shows that the proposed approach is applicable to problems with multiple non-smooth regions.

The same initial design of 25 experiments is used in order to compare the proposed approach with the SDO approach. Figure 5.17 shows the obtained metamodel using the SDO approach.

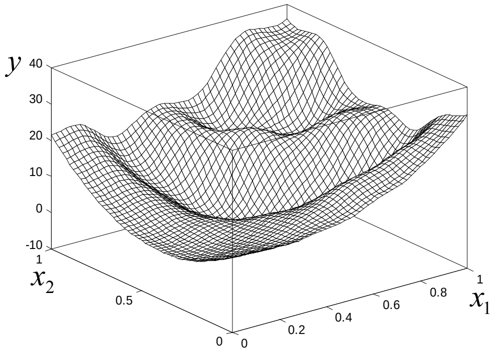
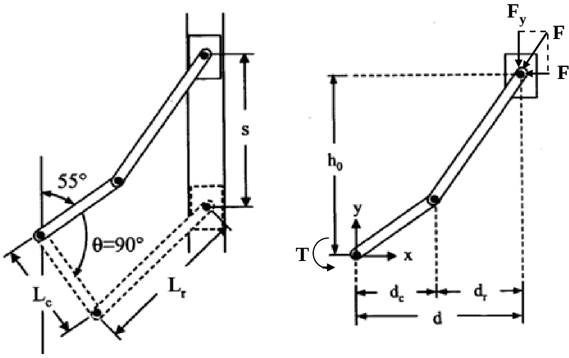


Figure 5.17 Metamodel for the SDO approach with RMSE = 0.024 and MAE = 0.233

Comparing Figure 5.16 and Figure 5.17, it can beobserved that the proposed approach and the SDO approach have a similar RMSE values, while a smaller MAE is obtained in the proposed approach. Again, it is observed that both the proposed approach and the SDO approach performed better than the maximum entropy design in terms of RMSE and MAE.

5.5.3 Linkage Example Two: Four-Input

The control value actuator linkage example used in this chapter is adapted from the examples in Gunawan (2004). This is an engineering example with explicit functions. The original problem from Gunawan (2004) has three design variables. We modified the problem and now it has four design variables. The modified problem is described as follows. The structure of a control value actuator linkage considered in this example is shown in Figure 5.18 as follows.



(a) (b)

Figure 5.18 A Control Value Actuator Linkage, courtesy from Gunawan (2004) and modified

The linkage shown above has a crank and a rod which are connected by a pin joint. And one end of the rod is also pinned to a slider. At the beginning, the crank is at a 55° angle from the vertical axis as shown in Figure 5.18(a). The dimensions of the linkage and the force applied on it are shown in Figure 5.18(b). The design objective is to maximize the average torque (*T*) at the end of the crank as it turns from 0° to 90°. It is assumed that the weight of the crank, rod, and slider is negligible. The design variables in the problem are the crank length (L_c), the rod length (L_r), the center distance (*d*), and the applied force (***F***). The crank length and the rod length are bounded between 0 and 10 inch, the center distance is bounded between 5 and 7 inch. In order to add non-smooth regions to the objective function, the applied force, which is not a design variable in the problem in Gunawan (2004), is considered as a design variable in this dissertation and is bounded between 0 and 1425.5 lbs with a variance coefficient *v*. Constraints of the problem are not considered in this example, since the emphases is to build a metamodel for the objective function *f*, which is defined in Eq. (5.10).

$$\begin{aligned}
&f = \frac{1}{10}\sum_{\theta=0}^{\pi/2} T(\theta) \\
&d_c = L_c \times \cos(\theta - \pi/4) \\
&d_r = d - d_c \\
&\beta = \cos^{-1}\left(\frac{d_r}{L_r}\right) \\
&h(\theta) = L_r \times \sin(\beta) - L_c \times \sin(\theta - \pi/4) \\
&F = 1425.5 \times v \\
&F_x = F \times \cos(\beta) \\
&F_y = F \times \sin(\beta) \\
&T(\theta) = d \times F_y - h(\theta) \times F_x
\end{aligned} \tag{5.10}$$

We begin with an initial design of 25 experiments. Then we apply the proposed approach to obtain 75 additional experiments. As in the previous examples, the maximum entropy design approach and the SDO approach were also applied to this example for comparisons. The obtained experiments are used to build metamodels for the objective function, and an additional testing sample of 1000 randomly selected to assess the overall metamodel accuracy by calculating RMSE and MAE. The metamodel accuracy, as discussed in Section 5.5.2, is critical in the offline metamodeling assisted optimization of deterministic simulation models. The results are shown in Table 5.2 as follows.

Table 5.2 RMSE and MAE for the Control Value Actuator Linkage Example

	Non-adaptive Approach	SDO Approach	Proposed Approach
RMSE	0.2166	0.1494	0.1528
MAE	0.1472	0.1508	0.1163

It can be observed from Table 5.2 that (i) the proposed approach provides a more accurate metamodel in which both RMSE (normalized) and MAE (normalized) are improved compared to the maximum entropy approach; (ii) the proposed approach and the SDO approach show similar RMSE values, and the proposed approach gives a smaller MAE.

Finally, note that the total number of experiments for metamodel building (e.g., 35 for example one, 50 for example two, and 100 for example three) is assumed to be the available number of function calls. In general, more experiments may be needed for problems with more design variables. This is because an adaptive DOE method is in essence a space filling method with emphases toward some regions but not other regions. Thus, as the dimension of the input space increases, more experiments will be required to fulfill the input space so that based on these experiments a metamodel with reasonable accuracy can be built.

5.6 Summary

The main contribution of the approach presented in this chapter is that we explicitly use the greedy principle for developing a proposed DOE method. As shown in the examples and test results, this greedy DOE method has outperformed a conventional (non-adaptive) DOE method in terms of the metamodel accuracy for the same number of experiments.

The proposed approach is adaptive and samples more experiments in regions of input space where the corresponding response is non-smooth. The approach starts with a set of initial design that is obtained using the maximum entropy design. Next, an initial metamodel is constructed by kriging. The remaining points are sampled one point at a time and then the kriging metamodel is updated accordingly. Using a modified leave-one-out strategy, the metamodel is used to find regions in the input space where the corresponding response is estimated to be non-smooth. The new point is placed in the non-smooth region so that the expected accuracy improvement of the metamodel is maximized. This process is repeated until a presumed number of simulation calls is reached. The proposed DOE approach has the ability to prevent clustering by adaptively determining a cluster threshold. This threshold is used to verify whether a newly selected point is sufficiently far from existing experiments.

This proposed approach is demonstrated with several numerical and engineering examples. The results from these examples show that for the same number of experiments and in terms of the metamodel accuracy, the proposed strategy outperforms both the maximum entropy design approach and the SDO approach. Note that this conclusion should be tempered by the assumption that the proposed approach is intended for problems where computational effort associated with simulations is much higher than that needed for sampling and metamodeling. If the computational cost for sampling and metamodeling cannot be ignored relative to that of the simulation model, then the proposed method can be less efficient than the conventional maximum entropy design approach.

In the next chapter, the research thrust four, the adaptive metamodeling for multi-response simulations, is presented.

Chapter 6 Metamodeling for Simulations with Multiple Responses

6.1 Introduction

Engineering design problems often have multiple outputs that need to be dealt with simultaneously. For example, manufacturers of motor vehicles need to ensure that the vehicles are safe (e.g., protect passengers in the case of an accident) via conducting crashworthiness testing. Several outputs can be obtained and analyzed as a result of an impact test, such as transmitted force and displacement. Also, such real-world engineering systems are complex and computing the objective/constraint functions and obtaining optimum solutions analytically are difficult and sometimes impossible. Therefore, computer simulation is frequently used in representing complex engineering systems for both responses evaluation and design optimization (Tekin and Sabuncuoglu, 2004). And these simulations can yield multiple responses for each execution.

In this chapter we present a new methodology for multi-response approximation that accounts for multiple responses available from a simulation model, both at the DOE stage and at the metamodel building stage. We label this new approach *dependent metamodeling*, as the model is built considering interdependencies between the different responses computed in a single simulation run. Unlike the conventional *independent* metamodeling approach that builds a single response metamodel for each response, the proposed *dependent* metamodeling approach builds a system of dependent metamodels based on all responses from the simulation.

In the proposed approach in this chapter, we use a combination of an extended maximum entropy design method (see Section 6.3.2) as our DOE approach and kriging-based method (see Section 6.3.1) for metamodeling. We define this approach as the proposed approach. On the other hand, the combination of the maximum entropy design method (recall Section 2.5) and the original kriging method (recall Section 2.6) are defined as the conventional approach. As a demonstration and for comparison, we applied both the proposed approach and the conventional approach to a numerical and an engineering example. The results show that the proposed approach outperforms the conventional approach in terms of the model accuracy for both examples.

The rest of the chapter is organized as follows. In Section 6.2, a brief review of multi-response deterministic simulations is outlined. Then in Section 6.3 the details of the proposed approach for multi-response deterministic simulations are given. Examples and corresponding results are discussed in Section 6.4. Finally, in Section 6.5 we summarize the main observations of the proposed metamodeling approach and discuss some future works.

A portion of this chapter was presented in Li et al. (2006a).

6.2 Multi-Response Deterministic Simulation

As discussed in Section 2.2, a computer simulation can be viewed as a black-box function that produces responses for given inputs. The dimension of the input and response of a simulation can be more than one. Especially when the response is a multi-dimension vector, a deterministic simulation is said to be a multi-response deterministic simulation, as the one represented in Figure 6.1. Such a simulation produces all the responses for an input in one simulation run.

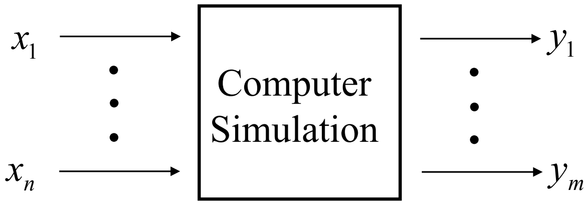


Figure 6.1 Multi-response deterministic simulation model

The deterministic simulation model represented in Figure 6.1 has a vector of input variables (design variables), denoted by $\boldsymbol{x} = (x_1, \ldots, x_n)$. The corresponding vector of responses, denoted by $\boldsymbol{y} = (y_1, \ldots, y_m)$. The universe $\boldsymbol{\Omega}^n$ that contains all design variable vectors is defined as the input space of the simulation, while the corresponding outputs from the computer simulation (see Figure 6.1) form the response space Λ^m. Thus, the deterministic relation between input and response vectors is:

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) \tag{6.1}$$

where $\boldsymbol{f}$: $\Omega^n \rightarrow \Lambda^m$ is a vector valued function that expresses possibly very complex relationship between design variables and responses. Note that $\boldsymbol{f}$ by itself has m components: $f_1, f_2, \ldots, f_m$, each of which, i.e., f_i, maps $\boldsymbol{x}$ to y_i, where $i = 1, 2, \ldots, m$. This system of functions, that are not known individually and explicitly, can be represented as in Figure 6.2.

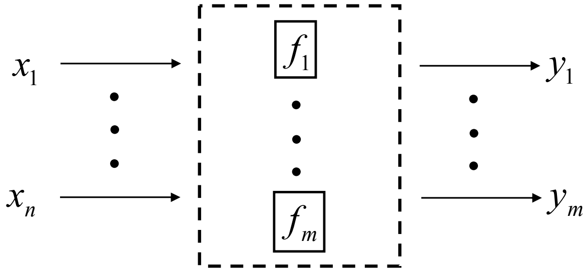


Figure 6.2 A system of functions in a multi-response simulation model

6.3 Metamodeling for Multi-Response Simulation

The goal of metamodeling is to estimate $\boldsymbol{f}(.)$, replacing it by the estimate $\hat{\boldsymbol{f}}(.)$, where the system of substitute $\hat{\boldsymbol{f}}$ is such that $\hat{\boldsymbol{y}} = \hat{\boldsymbol{f}}(\boldsymbol{x})$ approximates $\boldsymbol{y}$ as closely as possible. Note that $\hat{\boldsymbol{y}}$ is a vector defined as follows:

$$\hat{\boldsymbol{y}} = [\hat{y}_1, \ldots, \hat{y}_m] \tag{6.2}$$

where $\hat{y}_i$ is an estimate for a scalar response y_i, i = 1, 2, …, m. The conventional metamodeling approach creates an independent metamodel for each scalar response, i.e.,

$$\begin{aligned} \hat{y}_1 &= \hat{f}_1(x) \\ \hat{y}_2 &= \hat{f}_2(x) \\ &\vdots \\ \hat{y}_m &= \hat{f}_m(x) \end{aligned} \tag{6.3}$$

where $\hat{\boldsymbol{f}}(\boldsymbol{x})_i$, i = 1, 2, …, m, is a metamodel (e.g., kriging metamodel) for $\boldsymbol{f}(\boldsymbol{x})_i$.

Strictly speaking, in order to build a metamodel for each response, one needs to conduct experiments that only measure the ith response y_i. However, as discussed in Section 6.2, all responses (i.e., y_1, …, y_m) of a computer simulation are simultaneously obtained for any given input vector $\boldsymbol{x}$ = (x_1, …, x_n) after each run of the simulation. In building a metamodel for such a multi-response simulation, a lot of computation effort is going to be wasted if all the responses are treated separately. For example, m different DOEs are needed for such an approach, and the total number of function calls will be m times as many as function calls needed for each DOE. In practice, this large number of function calls is not desired, especially when the number of responses is increased. Figure 6.3 shows an example where each response has its own DOE.

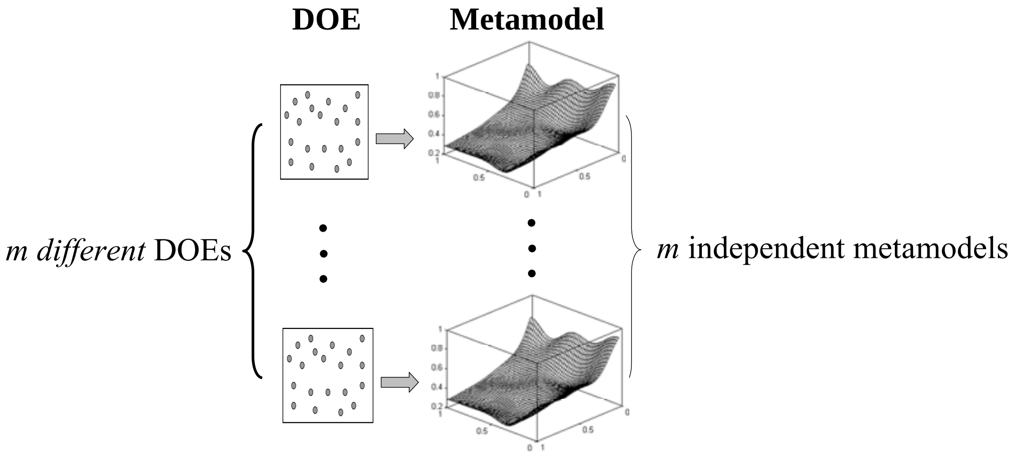


Figure 6.3 Metamodeling with different DOEs

Another way to address this problem is to use some sampling techniques independent from the response function, like the maximum entropy design approach, which is not dependent on the response function and always gives an almost uniform DOE based on a priori information and current available experiments. We can get a single DOE for all responses using this technique, but the accuracy of the resulting metamodels can be low since no information about response functions is used in these kinds of approaches. A representation for such approaches is given in Figure 6.4.

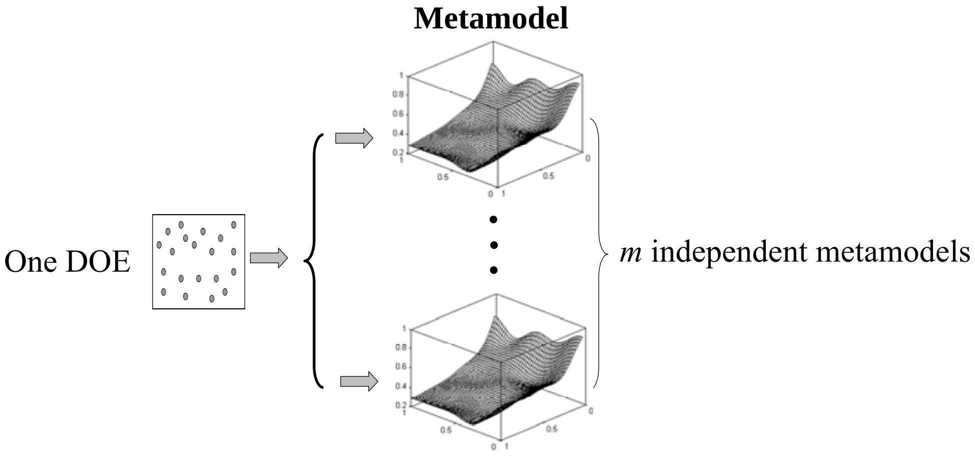


Figure 6.4 Metamodeling with one DOE

This latter approach is better than the approach of independent metamodeling with different DOEs in that it is expected to have a much less number of function calls.

In the next Section, a dependent metamodeling approach is developed in which the metamodels can be built with the same number of simulation runs as the approach of independent metamodeling with one DOE (see Figure 6.4), while achieving a good accuracy by using correlations among multiple responses.

6.3.1 Dependent Metamodeling

Dependent metamodeling is based on the idea that each single response is modeled as a function of all other responses, as well as the inputs. The goal of dependent metamodeling is to build a system of functions $\hat{g}_1(.)$, …, $\hat{g}_m(.)$ such that

$$\begin{aligned} \hat{y}_1 &= \hat{g}_1(\boldsymbol{x}, \hat{y}_2, \hat{y}_3, \ldots, \hat{y}_m) \\ \hat{y}_2 &= \hat{g}_2(\boldsymbol{x}, \hat{y}_1, \hat{y}_3, \ldots, \hat{y}_m) \\ &\vdots \\ \hat{y}_m &= \hat{g}_m(\boldsymbol{x}, \hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{m\text{-}1}) \end{aligned} \tag{6.4}$$

Each of the equations in Eq.(6.4) can be constructed using the kriging metamodeling, with estimates of responses included as part of the inputs for the equations, e.g., as shown in the first equation of Eq. (6.4): $\hat{y}_1$ is obtained based on $\boldsymbol{x}$ and $\hat{y}_2$, $\hat{y}_3$, …, $\hat{y}_m$. Note that, unlike Eq. (6.3), the dependent metamodels can not be solved independently to estimate a response. Instead, the equations in Eq. (6.4) must be solved simultaneously. Figure 6.5 shows an example of applying the dependent metamodeling and independent metamodeling to a simulation with two responses.

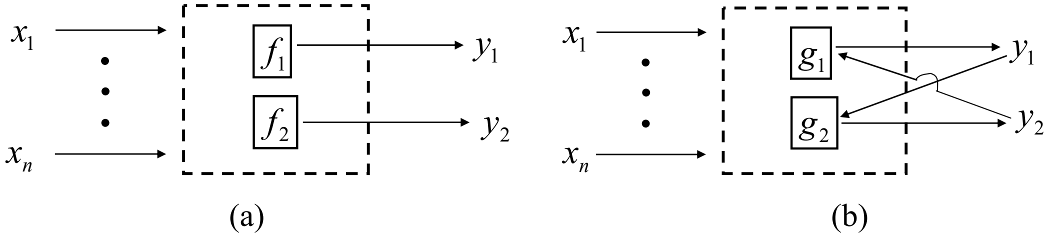


Figure 6.5 Scheme for (a) independent and (b) dependent metamodeling

The corresponding system of modeling equations for Figure 6.5(a) and Figure 6.5Figure 6.5(b) are shown in Eq. (6.5) and Eq. (6.6) separately as follows:

$$\begin{aligned}\hat{y}_1 &= \hat{f}_1(x)\\ \hat{y}_2 &= \hat{f}_2(x)\end{aligned} \tag{6.5}$$

$$\begin{aligned}\hat{y}_1 &= \hat{g}_1(x_1,\ \ldots,\ x_n,\ \hat{y}_2)\\ \hat{y}_2 &= \hat{g}_2(x_1,\ \ldots,\ x_n,\ \hat{y}_1)\end{aligned} \tag{6.6}$$

In the proposed approach, Eq. (6.4) is solved recursively in a two-step procedure: (i) In the first step, Eq. (6.3) is used to obtain an initial estimate for each response, i.e., $\hat{y}_1, \hat{y}_2,\ \ldots,\ \hat{y}_m$; and (ii) In the second step, these estimates are used in Eq. (6.4) to get new values for the estimates $\hat{y}_1, \hat{y}_2,\ \ldots,\ \hat{y}_m$, which are different from the initial estimates. Then, the second step is repeated until the values of estimates are converged, i.e., no significant change between two consecutive estimates is observed for every estimated response.

6.3.2. Extended Maximum Entropy Design

The maximum entropy design criterion is originally applied only in the $\boldsymbol{x}$ space, with the assumption that DOE is performed only in the $\boldsymbol{x}$ space. However, with regard to the dependent metamodeling approach, the maximum entropy design criterion must be extended to accommodate the new, larger input space (which includes both the input and response vectors) associated with Eq. (6.4), i.e.,

$$(\boldsymbol{x}^{i+1}, \boldsymbol{y}^{i+1}) = \arg\max\ H(\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^i, \boldsymbol{y}^1, \boldsymbol{y}^2, \ldots, \boldsymbol{y}^i;\ \boldsymbol{x}, \boldsymbol{y}) \qquad (6.7)$$

where in Eq. (6.7), H has the same definition as in Section 2.6, $\boldsymbol{x}$ and $\boldsymbol{y}$ are both vectors and the superscript for $\boldsymbol{x}$ or $\boldsymbol{y}$ refers to an experiment (or iteration) number. Note that the dimension of the search space in Eq. (6.7) differs from that for a conventional maximum entropy design method as defined in Eq. (2.12). Solutions to (6.7) are vectors $(x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_m)$ of dimension $n + m$. Furthermore, since $\boldsymbol{y} = (y_1, y_2, \ldots, y_m)$ is in fact a function of $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ (although the relationship is not known a priori), not every combination $(\boldsymbol{x}, \boldsymbol{y}) = (x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_m)$ is *consistent* as defined in the following.

Definition: A vector $(\boldsymbol{x}, \boldsymbol{y}) = (x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_m) \in \boldsymbol{\Omega}^{n+m}$ is said to be consistent if the simulation yields $(y_1, y_2, \ldots, y_m)$ as the response for the input $(x_1, x_2, \ldots, x_n)$. In other words, in an n-input, m-response deterministic simulation model (i.e., Figure 6.2), $(x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_m)$ is consistent if $y_j = f_j(\boldsymbol{x}), j = 1, \ldots, m$.

Clearly, there is no guarantee that the solution to Eq. (6.7) is consistent if we solve it directly as an unconstrained optimization problem. A measure of the consistency of an arbitrary vector $(x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_m)$ is expressed by means of the functions δ_j defined as

$$\delta_j\,(\boldsymbol{x},\boldsymbol{y}) = |\, y_j - f_j(\boldsymbol{x})| \qquad (6.8)$$

for $j = 1,\ldots, m$. However, since an exact evaluation of the functions $f_j(\boldsymbol{x})$ is not available, functions $\delta_j\,(\boldsymbol{x}, \boldsymbol{y})$ in Eq. (6.8) must be estimated. This can be done using the metamodeling functions $\hat{g}_j(\boldsymbol{x})$ (as defined in Eq. (6.4)) to replace $f_j(\boldsymbol{x})$, thus,

$$\begin{aligned}
&\hat{\delta}_1(\boldsymbol{x},\boldsymbol{y}) = \left|\hat{g}_1^k(x, y_2, y_3, \ldots, y_m) - y_1\right| \\
&\hat{\delta}_2(\boldsymbol{x},\boldsymbol{y}) = \left|\hat{g}_2^k(x, y_1, y_3, \ldots, y_m) - y_2\right| \\
&\vdots \\
&\hat{\delta}_m(\boldsymbol{x},\boldsymbol{y}) = \left|\hat{g}_m^k(x, y_1, y_2, \ldots, y_{m-1}) - y_m\right|
\end{aligned} \tag{6.9}$$

where $\hat{g}_j^k$ is the *k*th generation metamodeling for $f_j(\boldsymbol{x})$. Therefore $(x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_m)$ is estimated to be consistent if $\hat{\delta}_1(\boldsymbol{x},\boldsymbol{y}) = \hat{\delta}_2(\boldsymbol{x},\boldsymbol{y}) = \ldots = \hat{\delta}_m(\boldsymbol{x},\boldsymbol{y}) = 0$ according to the current generation metamodel. In order to account for the consistency condition, the conventional maximum entropy design criterion defined in Eq. (2.12) is replaced by a constrained maximization problem: Find $(\boldsymbol{x}^{i+1}, \boldsymbol{y}^{i+1})$ such that

$$\begin{aligned}
&\text{maximize } \; H(\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^i, \boldsymbol{y}^1, \boldsymbol{y}^2, \ldots, \boldsymbol{y}^i;\; \boldsymbol{x}, \boldsymbol{y}) \\
&\text{subject to } \; \hat{\delta}_j(\boldsymbol{x},\boldsymbol{y}) \le \varepsilon,\; j = 1, \ldots, m
\end{aligned} \tag{6.10}$$

where ε is a tolerance parameter, that is a small positive constant that can be determined based on the density of the normalized input space, i.e., should be less than the distance between any two points in the input space; *i* is the number of experiments already in the design. This design criterion is used along with the system of dependent metamodels (i.e., Eq. (6.4)) to build metamodels for a multi-response deterministic simulation. Satisfying the consistency constraint in Eq. (6.10) is crucial to the success of the strategy. However, computational effort for satisfying the consistency constraint can be ignored, given the assumption that the computational cost for each simulation run is overwhelming.

In the above mentioned proposed approach, the space on which the maximum entropy design is used has two characteristics that make it different from the conventional maximum entropy approach. First, the input space of the proposed approach has a higher dimension than that of input space of the simulation because the responses of the simulation are also treated as part of the input space. Second, the entropy maximization problem in the proposed approach is constrained while the conventional maximum entropy design is unconstrained. Therefore, experiments obtained from the proposed approach will be different from the conventional approach since they are designed in a higher-dimension, mapped to the input space of the simulation, and forced to be consistent using the inequality constraints as defined in Eq. (6.10).

6.3.3 Steps for Dependent Metamodeling

The proposed approach, which includes a metamodeling approach and a sampling method, can be represented in Figure 6.6, and detailed steps as given next.

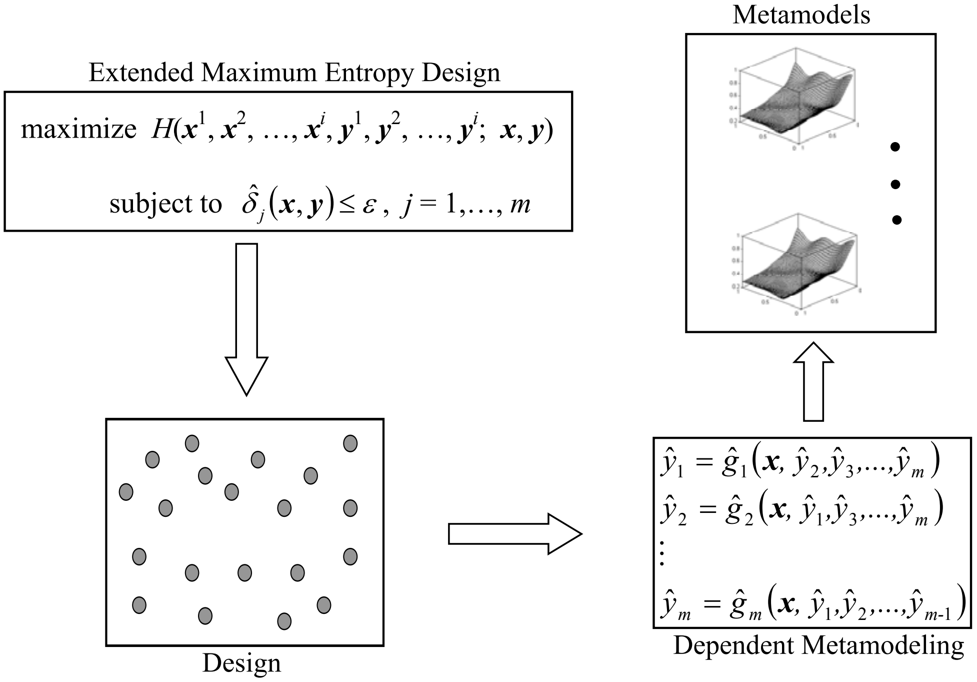

Metamodels
Extended Maximum Entropy Design
maximize $H(\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^i, \boldsymbol{y}^1, \boldsymbol{y}^2, \ldots, \boldsymbol{y}^i;\ \boldsymbol{x}, \boldsymbol{y})$
subject to $\hat{\delta}_j(\boldsymbol{x}, \boldsymbol{y}) \leq \varepsilon$, $j = 1, \ldots, m$
$\hat{y}_1 = \hat{g}_1(\boldsymbol{x}, \hat{y}_2, \hat{y}_3, \ldots, \hat{y}_m)$
$\hat{y}_2 = \hat{g}_2(\boldsymbol{x}, \hat{y}_1, \hat{y}_3, \ldots, \hat{y}_m)$
$\vdots$
$\hat{y}_m = \hat{g}_m(\boldsymbol{x}, \hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{m\text{-}1})$
Dependent Metamodeling
Design

Figure 6.6 Overall framework of the proposed approach

Normalization: We assume that the region of interest where experiments are to be designed (the input design space) is bounded by upper and lower limits on each of the design variables and that normalization is possible so that the design space is bounded within $[0, 1]^n$. Similarly, we also assume that the response space can be normalized using an estimate of the lower and upper limits of each response. In general, this is not a severe assumption, as the estimates need not be tight and may be simply set to values that result in intervals large enough to include all values of responses. This results in a normalized response space contained in $[0, 1]^m$. After normalization, we can use the constrained maximization problem as defined in Eq. (6.10) to obtain new experiments at a rate of one point per iteration, as explained in the following steps.

Initial Design: The process starts with a randomly selected single initial experiment. Note that the proposed method may become too complicated and lose some of its appeal if an adaptive design criterion is used to multiple points are selected initially. Fortunately, in general, after enough points are added, the accuracy and efficiency of the proposed method is not affected significantly by how well the initial points are distributed. Nevertheless, it must be acknowledged that in problems with a very small number of experiments, the choice of the initial point can become somewhat significant, but the exact effect only depends on the properties of the unknown response and cannot be predicted if it is assumed that no information from the response is available initially.

Stopping Criterion: The number of available simulation runs is used as the stopping criterion, determined a priori from the available computation resources, under the assumption that simulations account for the bulk of the computational effort. Model accuracy is not used as a stopping criterion in this chapter simply to avoid extra simulation runs. The proposed algorithm is shown in Figure 6.7 and the detailed steps are also given below the figure.

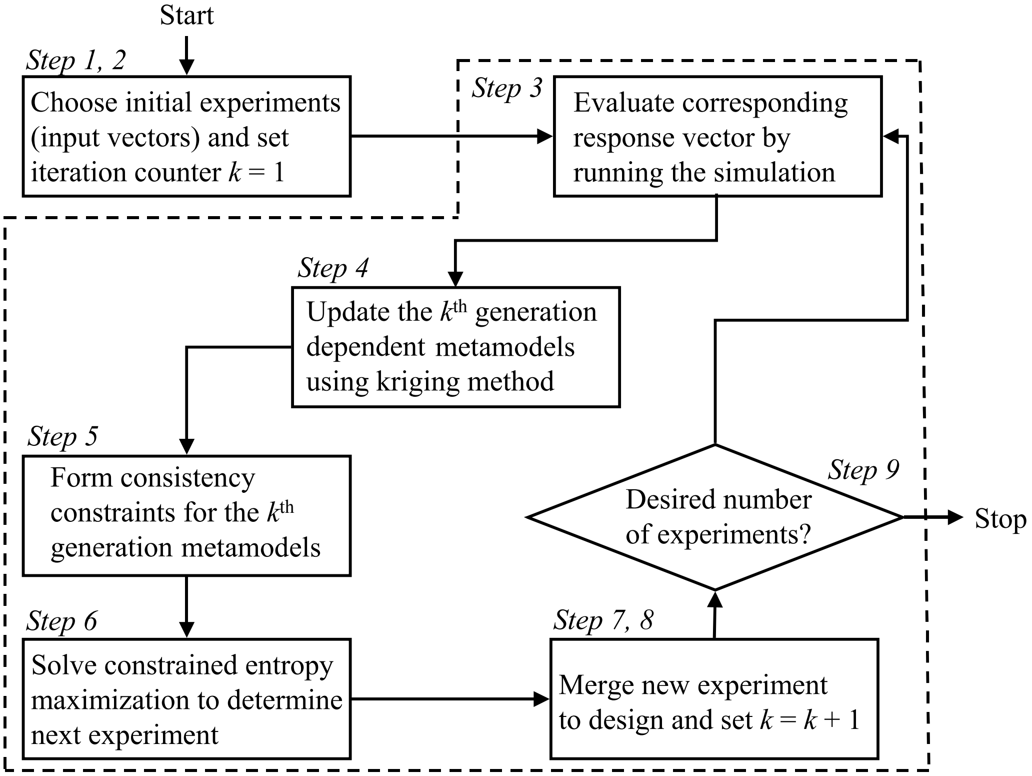

Start
Step 1, 2
Choose initial experiments (input vectors) and set iteration counter k = 1
Step 3
Evaluate corresponding response vector by running the simulation
Step 4
Update the kth generation dependent metamodels using kriging method
Step 5
Form consistency constraints for the kth generation metamodels
Step 6
Solve constrained entropy maximization to determine next experiment
Step 7, 8
Merge new experiment to design and set k = k + 1
Step 9
Desired number of experiments?
Stop

Figure 6.7 Flowchart for dependent metamodeling

<u>Step by step description of the algorithm:</u>

<u>Step 1</u>: A single initial experiment $\boldsymbol{x}^1$ is chosen randomly from within the normalized design space. The design $\boldsymbol{D}$ is initialized, $\boldsymbol{D} = \{\boldsymbol{x}^1\}$.

<u>Step 2</u>: Define k as the iteration counter and set $k = 1$.

<u>Step 3</u>: The response vector for the current design $\boldsymbol{D}$ ($\boldsymbol{D} = \{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^k\}$) is obtained by running the multi-input, multi-response simulation model.

<u>Step 4</u>: The kriging based dependent metamodeling is used to update the kth generation dependent metamodels (recall Section 6.3.1).

<u>Step 5</u>: The consistency measures $\hat{\delta}_j$ are built using the kth generation dependent

metamodels $\hat{g}_j^k$ for each input vector in a normalized design space.

Step 6: The constrained maximum entropy optimization problem defined in Eq. (6.10) is solved to obtain the next input vector which maximizes entropy in the consistent input-response space (i.e., the space of vectors that are feasible according to the consistency constraints of the problem in Eq. (6.10)).

Step 7: The newly selected experiment $\boldsymbol{x}^{k+1}$ (i.e., a single experiment for each stage) is merged into design $\boldsymbol{D}$.

Step 8: Set $k = k + 1$.

Step 9: Stop if the stopping criterion is achieved (e.g., desired number of experiments in $\boldsymbol{D}$ is reached). Otherwise, go to Step 3.

The above algorithm is also shown in a flowchart as follows:

6.4 Examples and Results

In this section two examples are used to test the proposed approach. In each example, kriging models are constructed using both the conventional and the proposed approach. A testing sample with randomly selected experiments is used to estimate the model accuracy. The root mean squared error (RMSE) of the metamodels is used to evaluate the model accuracy.

6.4.1 Numerical Example

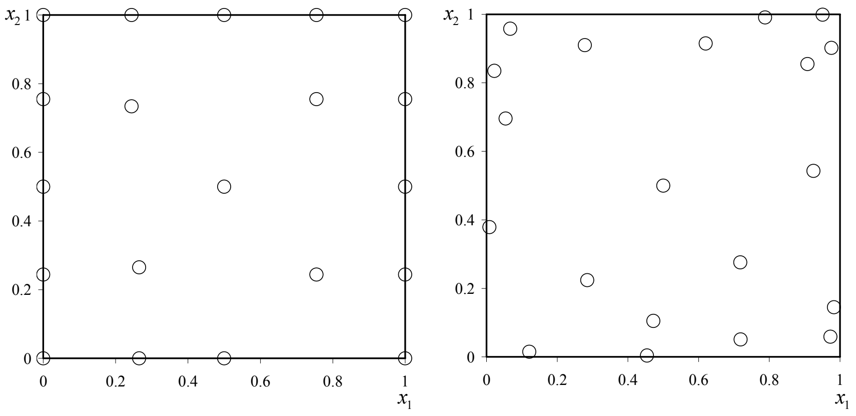
Consider the following two-input (i.e., n = 2) two-response (i.e., m = 2) explicit numerical problem (this is a modification of a single-response problem, first proposed by Farhang-Mehr and Azarm (2002, 2005) as a challenging test problem with varying response behavior and many local optima :

$$
\begin{aligned}
y_1 &= x_1 x_2 + 0.05 \\
y_2 &= x_1 \{(1 - e^{-2\sqrt{x_2}}) + 6x_2 e^{-7x_2} \sin(10x_2) - 0.2e^{-2000(x_2 - 0.25)^2} \\
&\quad + 60 \min(0, |x_2 - 0.14| - 0.08)^2 [\ln(x_2 + 0.2) + 1.5 \sin^2(85x_2)]\}
\end{aligned}
\tag{6.11}
$$

Table 6.1 includes a tabulation of 20 experiments (listed under "Conventional Approach" in Table 6.1) obtained using the maximum entropy design approach, which is also graphed in Figure 6.8(a). The order in which the experiments were obtained is shown in Table 6.1. Independent metamodels are then created separately using the kriging method for each response. Table 6.1 also shows a tabulation of a different design of 20 experiments (listed under "Proposed Approach" in Table 6.1) obtained using the new multi-response approach of this chapter (see Figure 6.8(b)).

Table 6.1 Design of 20 experiments using conventional and proposed approaches

(1): Conventional Approach (See Figure 6.8(a))			(2): Proposed Approach (See Figure 6.8(b))		
Experiment	x_1	x_2	Experiment	x_1	x_2
1	0.500	0.500	1	0.500	0.500
2	0.000	0.000	2	0.951	0.999
3	0.000	1.000	3	0.973	0.059
4	1.000	0.000	4	0.121	0.015
5	1.000	1.000	5	0.067	0.958
6	0.000	0.500	6	0.925	0.543
7	1.000	0.500	7	0.008	0.379
8	0.500	1.000	8	0.620	0.915
9	0.500	0.000	9	0.454	0.004
10	0.244	0.734	10	0.054	0.696
11	0.265	0.265	11	0.908	0.855
12	0.755	0.755	12	0.278	0.910
13	0.755	0.244	13	0.718	0.276
14	0.755	1.000	14	0.285	0.224
15	0.000	0.755	15	0.983	0.145
16	0.265	0.000	16	0.472	0.105
17	1.000	0.755	17	0.788	0.991
18	0.000	0.244	18	0.719	0.051
19	0.244	1.000	19	0.022	0.835
20	1.000	0.244	20	0.976	0.902

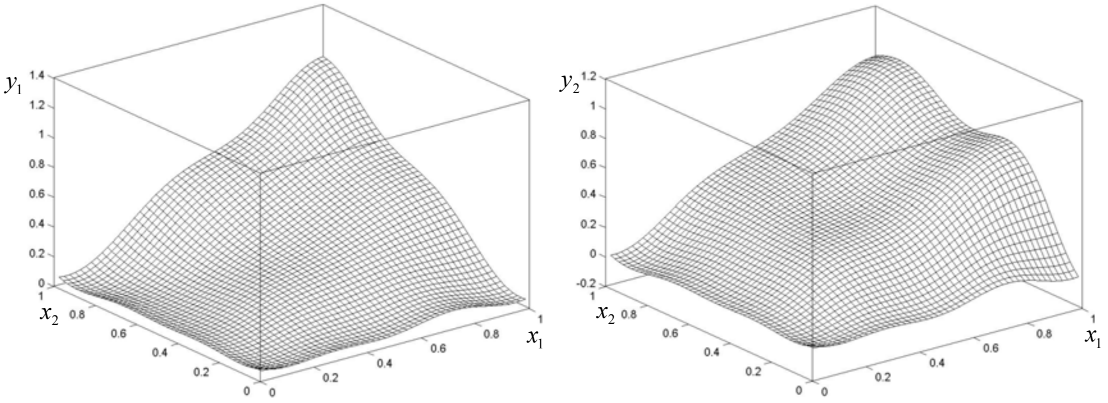


(a) (b)

Figure 6.8 Design of 20 experiments using (a) conventional independent approach, and (b) proposed dependent approach

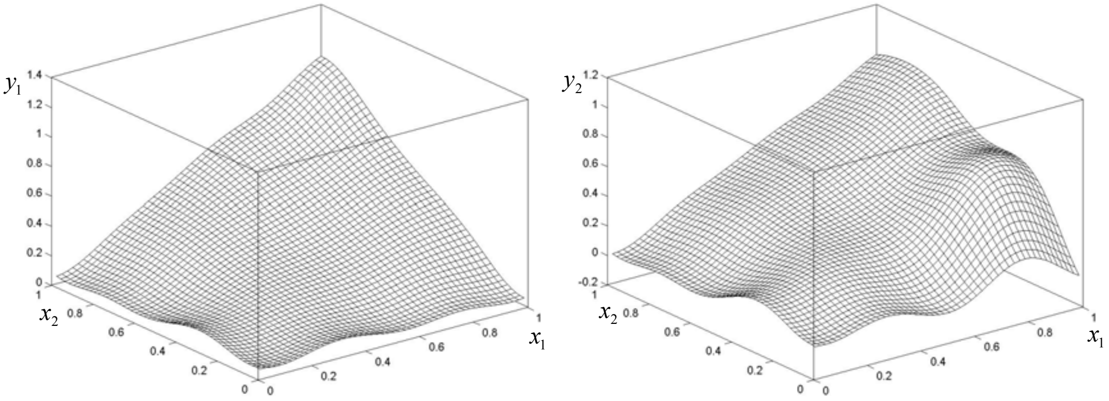
In the approach in this chapter, we use y_2 as well as x_1 and x_2 as the input vector to estimate y_1, and y_1 as well as x_1 and x_2 as the input vector to estimate y_2. Unlike the conventional approach, the distribution of the points in the new approach is asymmetric (Figure 6.8(a) is almost symmetric with emphasis on the boundaries). It means that the information from the other response has some effects in the selection of experiments and this additional information may help to build a more accurate metamodel, as observed in the RMSE results discussed in the next paragraph.

Metamodels in both Figure 6.9 and Figure 6.10 are obtained after normalization. Figure 6.9 shows the metamodels and RMSE values obtained using the conventional maximum entropy approach. Figure 6.10 shows the metamodels and RMSE values obtained from the proposed approach of this chapter using the same number of experiments (20 simulation runs). To estimate the accuracy of the obtained metamodels (from both approaches), we used an additional testing sample of 1000 randomly selected experiments and obtained the RMSE for each metamodel. Comparing the metamodels in Figure 6.9 with the corresponding metamodels in Figure 6.10, it can be observed that the proposed approach provides more accurate approximations (i.e., lower RMSE values) for both responses.



(a) (b)

Figure 6.9 Metamodels from the conventional approach for (a) y_1 with $\text{RMSE}(y_1)\sim 0.28$, and (b) y_2 with $\text{RMSE}(y_2)\sim 0.39$



(a) (b)

Figure 6.10 Metamodels from the dependent metamodeling approach for (a) y_1 with $\text{RMSE}(y_1)\sim 0.05$, and (b) y_2 with $\text{RMSE}(y_2)\sim 0.11$

6.4.2 Engineering example - Crashworthiness design simulation

By most measures, simulation of a crash event involving a typical vehicle is a computationally intensive task. A detailed computer model of a passenger vehicle typically involves 10^5 – 10^6 degrees of freedom and one performance evaluation may require many hours or days of computer time. The physical phenomena involved are highly non-linear and the simulated responses tend to be quite ill-behaved. Because of this complexity, metamodeling of such simulations are crucial for crashworthiness design. A detailed multi-purpose finite element model of a 1994 Chevrolet C-2500 pick-up truck has been developed at the National Crash Analysis Center at George Washington University (Bedewi et al., 1996). A representation of the rail assembly is shown in Figure 6.11 as follows.

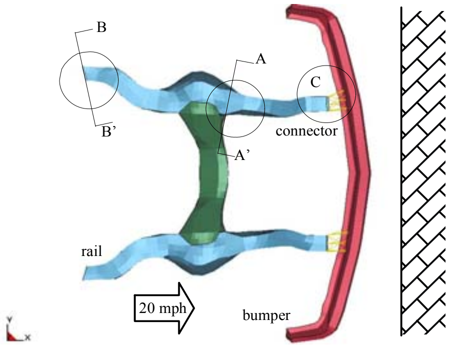


Figure 6.11 The bumper-rail assembly

The assembly consists of the bumper, the left and right rails, and the cross rail connector. The rail mountings, which connect the bumper and the rails, were replaced with connectors modeled by beam elements whose purpose is simply to engage/disengage the rail and the bumper. Lumped masses are attached at the rear end of the rails, at section B-B'. As shown in Figure 6.11, the assembly is moving forward to the right at a speed of 20 mph when it hits a rigid wall. The stress analysis of this assembly was performed using a standard finite element package for large deformation and impact analysis.

In designing for enhanced crashworthiness, the objective is to improve the protection of the passenger, e.g., by controlling the accelerations experienced by the passenger and the deformation of the structure in the immediate vicinity of the passenger. In our case this is accomplished indirectly by (1) the maximum force, *F*(*t*), transmitted through the rail (measured at section A-A'); and (2) the maximum (X-)

displacement $D(t)$ of the section at B-B'. The simulation model therefore, computes two responses:

$$y_1 = \max_{t>0} \left| F(t) \right| \tag{6.12}$$

$$y_2 = \max_{t>0} \left| D(t) \right| \tag{6.13}$$

The simulation model calculates both of these responses in one run. The simulation run was performed by Dr. Diaz and his students at the Michigan State University (see Li et al., 2006a), and each run takes about 15 minutes on a Sun Ultra 80 workstation. Because of the non-linear and multi-response nature of this model, it is a particularly interesting case study for the proposed approach of this chapter.

In Table 6.2, column 1 tabulates 36 experiments obtained using the conventional maximum entropy approach, which is also graphed in Figure 6.12(a). Table 6.2 also shows a tabulation of a different design of 36 experiments (listed under "Proposed Approach" in Table 6.2) obtained using the new multi-response approach of this chapter. Figure 6.12(b) shows the obtained experiments in the input space. The comparison of obtained designs is similar to that in the first example.

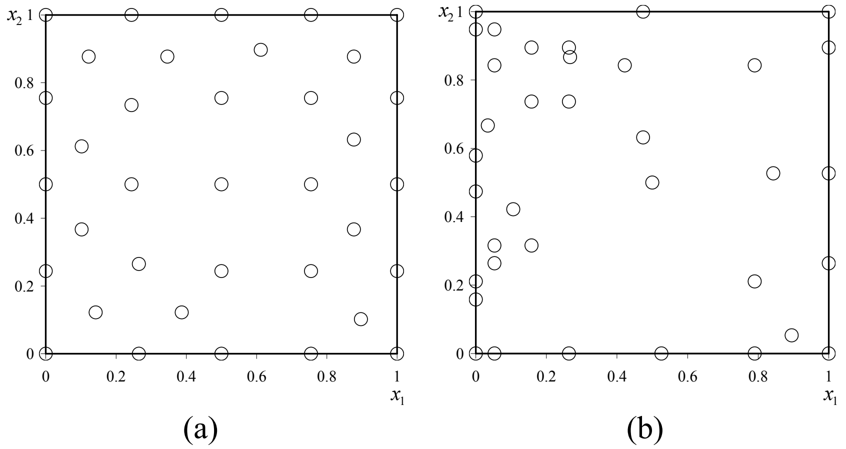


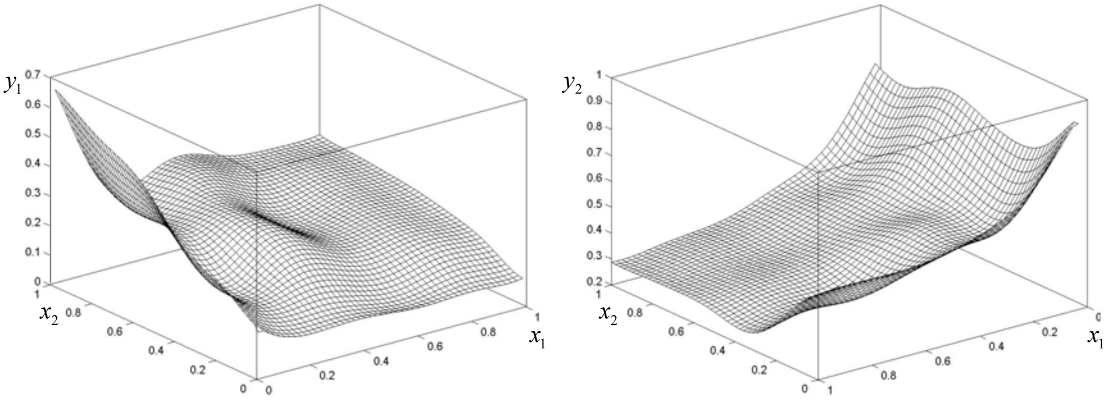
Figure 6.12 Design of 36 experiments using (a) conventional independent approach, and (b) proposed dependent approach

Table 6.2 Design of 36 experiments for crashworthiness design example

(1): Conventional Approach (See Figure 6.12(a))			(2): Proposed Approach (See Figure 6.12(b))		
Experiment	Force	Displacement	Experiment	Force	Displacement
1	1.000	1.000	1	0.500	0.500
2	0.000	0.000	2	0.000	0.000
3	0.000	1.000	3	1.000	1.000
4	1.000	0.000	4	1.000	0.000
5	0.500	0.500	5	0.000	1.000
6	0.000	0.500	6	0.474	1.000
7	1.000	0.500	7	0.000	0.473
8	0.500	1.000	8	0.526	0.000
9	0.500	0.000	9	1.000	0.421
10	0.244	0.734	10	0.842	0.684
11	0.265	0.265	11	0.210	0.789
12	0.755	0.755	12	0.210	0.157
13	0.755	0.244	13	0.000	0.736
14	0.755	1.000	14	0.157	0.473
15	0.000	0.755	15	0.578	0.157
16	0.265	0.000	16	0.105	0.947
17	1.000	0.755	17	0.947	0.157
18	0.000	0.244	18	0.842	0.105
19	0.244	1.000	19	0.210	1.000
20	1.000	0.244	20	0.842	0.263
21	0.755	0.000	21	0.736	1.000
22	0.500	0.755	22	0.000	0.105
23	0.500	0.244	23	0.947	0.052
24	0.755	0.500	24	1.000	0.052
25	0.244	0.500	25	1.000	0.526
26	0.142	0.122	26	0.736	0.105
27	0.122	0.877	27	0.733	0.133
28	0.877	0.877	28	0.947	1.000
29	0.877	0.632	29	0.894	0.578
30	0.387	0.122	30	1.000	0.789
31	0.346	0.877	31	0.947	0.684
32	0.897	0.102	32	0.736	0.263
33	0.877	0.367	33	0.966	0.333
34	0.102	0.367	34	0.526	0.368
35	0.612	0.897	35	1.000	0.842
36	0.102	0.612	36	0.947	0.736

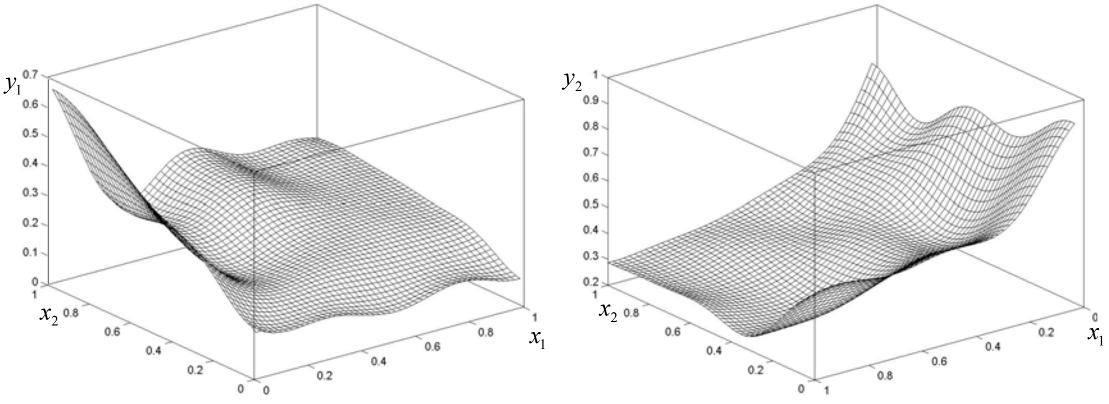
Figure 6.13 and Figure 6.14 shows the resulting metamodels and RMSE values from the conventional and the proposed approach using the same number of experiments (36 simulation runs). Note that force and displacement values are normalized in Figure 6.13 and Figure 6.14.

In Figure 6.13 and Figure 6.14, the RMSE values of the obtained metamodels are estimated using a random sample of 40 experiments. Comparing the metamodels in Figure 6.13 with the corresponding metamodels in Figure 6.14, it can be observed that the proposed approach provides more accurate approximations for both responses (i.e., lower RMSE values).



(a) (b)

Figure 6.13 Metamodels from the conventional approach for (a) maximum force y_1 with $\text{RMSE}(y_1)\sim 0.24$, and (b) maximum displacement y_2 with $\text{RMSE}(y_2)\sim 0.49$



(a) (b)

Figure 6.14 Metamodels from the dependent metamodeling approach for (a) maximum force y_1 with $\text{RMSE}(y_1)\sim 0.18$, and (b) maximum displacement y_2 with $\text{RMSE}(y_2)\sim 0.23$

6.5 Summary

A new methodology for offline approximation of multi-response simulations is proposed in this chapter, one that creates and updates mutually dependent metamodels. These metamodels are created assuming that other response values from the model can be used as inputs, creating a system of dependent metamodels that must be solved simultaneously. Unlike traditional (independent) metamodeling techniques, the new approach takes advantage of the cross-correlation among multiple responses to improve the accuracy of the obtained metamodel (with the same number of experiments), and to improve the efficiency for selection of new sample locations using the information present in all of the observations.

The two examples in this chapter illustrate that, for the same number of simulation runs and in terms of the accuracy of the obtained metamodels, the new approach outperforms a conventional method by a good margin. We should emphasize that this conclusion must be tempered by the fact that the proposed approach is intended for problems where computational costs associated with simulations are overwhelming. If the computational cost for internal computations cannot be ignored relative to that of the simulation model, then the proposed method is less efficient than the conventional approach.

In the proposed approach, the experiments are generated at a rate of one point for each iteration since the fully sequential design is the most natural way for computer simulations (e.g., Jin et al., 2002; Sacks et al.; 1989). Furthermore, as mentioned before, the assumption is that the computational effort for each simulation is much more intensive in comparison to that required for metamodeling and DOE. Although the proposed approach is developed for generating one point at a time, it can also be readily extended to generating several points at a time by conducting each corresponding DOE on one processor, and use the rest of the processors to evaluate the experiments by running the simulation (the main source of computational burden) separately at each processor.

Finally, the number of available simulation runs and the desired model accuracy are often used as the stopping criterion in a sequential approximation process. In this chapter, the number of available simulation runs is used as the stopping criterion. The model accuracy in terms of RMSE, on the other hand, can be obtained using two different methods. One method is to use an additional testing sample with a set of randomly selected experiments to calculate the RMSE value, which is unbiased but requires extra simulation runs. In practice, these extra simulation runs are not desirable since the main purpose for approximation is to use fewer simulation runs. Another method that is used to estimate the model accuracy is cross-validation (e.g., Kohavi 1995), which does not require an additional testing sample. However, the cross-validation method can be biased because the sample for testing is not selected randomly. One possible future direction calls for an un-biased and more efficient metamodel accuracy assessment method.

In the next chapter, the conclusions of this dissertation are presented.

Chapter 7 Conclusions

This dissertation presented the approaches and results for four research thrusts in the context of approximation assisted population based multi-objective optimization. These were: 1) Fitness Estimation in Online Metamodeling Research Thrust, wherein an online kriging based fitness estimation approach in online for population based optimization methods was developed; 2) Adaptive DOE in Online Metamodeling Research Thrust, wherein an adaptive DOE method that can identify predicted optima was developed; 3) Adaptive DOE in Offline Metamodeling Research Thrust, wherein a greedy algorithm for adaptive DOE was developed that can sample more experiments in non-smooth regions, and thus improves the metamodel accuracy; and 4) Dependent Metamodeling Research Thrust, wherein a metamodeling approach was developed that accounts for the correlation among the responses from a simulation with multiple responses.

This last chapter is organized as follows. In Section 7.1, the four research presented in the previous four chapters are summarized. After that, some conclusions about the presented approaches are given in Section 7.2. In Section 7.3, the main contributions of this dissertation are outlined. Finally, some future research directions are discussed in Section 7.4.

7.1 Summary

In the following, a summary of the four research thrusts is provided.

- *Fitness Estimation in Online Metamodeling Research Thrust*: As presented in Chapter 3, the kriging metamodeling assisted fitness evaluation approach in a population based optimization method is developed as a new online metamodeling technique. The main idea is that some of the points are evaluated online using kriging metamodeling instead of the original simulation model in the optimization procedure. In this way, the number of function calls can be reduced in each generation in a population based optimization approach. The identified points to be estimated using the kriging metamodel are those that the corresponding estimated response would not change the estimated dominance status in the objective space for the current generation. If the dominance status is estimated to be changed, then the simulation is used for evaluating that point. Otherwise, the metamodel is used.

- *Adaptive DOE in Online Metamodeling Research Thrust*: As presented in Chapter 4, a new adaptive DOE method is developed. In the proposed DOE method, the predicted optima are identified, from which some points are selected and added to the next population in a population based optimization approach. It is shown that the number of generation needed for convergence is reduced. Also, this new adaptive DOE method is integrated with the kriging metamodeling assisted fitness evaluation approach developed in Chapter 3 in order to reduce the overall number of function calls even more when compared to MOGA or even K-MOGA.
- *Adaptive DOE in Offline Metamodeling Research Thrust*: As presented in Chapter 5, a greedy algorithm based DOE approach is developed for offline metamodeling. The proposed approach is denoted as an *ACcumulative Error* (ACE) approach for DOE. ACE is adaptive and samples more points in regions of input space where the response is non-smooth. The non-smooth regions are critical to identify and sample in order to improve the metamodel accuracy.

- *Dependent Metamodeling Research Thrust*: As presented in Chapter 6, a new approach to metamodeling is introduced whereby a sequential technique is used to construct and simultaneously update mutually dependent metamodels for high-fidelity simulations with multiple responses. This new approach is a dependent metamodeling approach. Unlike conventional approaches which produce a single metamodel for each response independently, this new dependent metamodeling approach uses the correlation among different simulation outputs (or responses) in the construction of the metamodel. These dependent metamodels are solved as a system of equations to estimate individual responses simultaneously.

7.2 Conclusions

In the following subsections, concluding remarks are given for each research thrust.

7.2.1 Fitness Estimation in Online Metamodeling Research Thrust

In this research thrust, the concept of *MMD* is introduced to distinguish between an estimated dominated and non-dominated set of points in a population. The relation between *MMD* and the uncertainty associated with the estimated responses is derived. This relation is used to objectively determine the condition for using the simulation or metamodel for evaluating a point, so that the number of function calls can be saved as the response for some points are estimated using the metamodel.

The kriging assisted fitness evaluation approach presented in Chapter 3 was applied to several engineering and numerical test problems. These test problems have different degrees of difficulty and characteristics in terms of the design variables and constraints. For instance, some test examples have integer design variables, while others have continuous design variables. Moreover, the shapes of the Pareto frontiers are quite different as shown in Chapter 3. We also used a conventional MOGA as a baseline optimization approach. For these test examples and on the average, a MOGA with the kriging assisted fitness evaluation (i.e., K-MOGA) reduces by a factor of two the number of function calls required compared to a conventional MOGA.

In the proposed approach, the kriging metamodel is updated in each generation in the optimization process. As a result, the metamodel accuracy is improved gradually. That is, those points that are migrated from previous generations with incorrectly estimated error are likely to be removed from the population due to a more accurate kriging metamodel. The side effect of such migrated points can be diminished when the kriging metamodels are updated adaptively in consecutive generations. In essence, the proposed approach has a self-correcting mechanism in terms of identifying good points for kriging metamodeling.

There are some shortcomings in the proposed approach. One is that the relation between the kriging's predicted error and *MMD* is devised based on a worst case scenario and thus the proposed approach can be considered to be conservative. Another is that the proposed approach requires extra computation for *MMD* calculation and metamodel construction. In this regard, it is often assumed that the computational cost for running the simulation is much more than that for the overhead in the proposed approach and thus the extra computational effort is neglected.

7.2.2 Adaptive DOE in Online Metamodeling Research Thrust

In Chapter 4, a new adaptive DOE method is developed to identify the predicted optima, and from which some points are selected as part of the next population. These points are defined as DOE points, and are in addition to the points from inheritance, crossover and mutation used in the next population. The DOE method is combined with the online metamodeling approach from Chapter 3 and together with a population based optimization method, such as MOGA, to reduce the number of function calls.

The same set of test problems (except for the engineering examples) from Chapter 3 are also used in Chapter 4 to demonstrate the applicability of the approach in this research thrust. Also, the same conventional MOGA is used for comparison. It is observed that on the average, the number of function calls is reduced by about 70%. In other words, the use of the proposed DOE method contributes to an additional 20% saving in the number of function calls compared to K-MOGA. This extra saving, as we discussed in Chapter 4, is due to the reduction in the number generations for convergence. Moreover, the proposed DOE approach is applicable to MOGA and other population based optimization methods.

One shortcoming of the proposed DOE method is that the predicted optima is obtained by running an optimizer in the metamodel, which is another computational overhead in the improved optimization framework in addition to that from the fitness estimation approach. However, again, as in the same previous assumption, the computational cost for running the simulation is assumed to be much more than that from the overhead cost and thus the overhead can be neglected.

7.2.3 Adaptive DOE in Offline Metamodeling Research Thrust

In this research thrust, as discussed in Chapter 5, a new adaptive DOE method is developed in which: i) a modified leave-one-out method is used to identify non-smooth regions, and ii) the greedy algorithm is used to select new experiments in the non-smooth regions.

Several test problems are used to show the strength of the proposed greedy based adaptive DOE approach. These problems have single or multiple inputs. Also, the response surfaces (for problems with one or two inputs) have different degree of complexity. It is observed that the proposed DOE approach outperforms a conventional DOE method in terms of the metamodel accuracy using the same number of function calls for the test examples.

One shortcoming of the proposed DOE approach is that we used an ad-hoc approach to determine how many of experiments should be used for initial design, before the approach begins and how many during the application method. This aspect will need to be further investigated although some initial results are presented in Chapter 5.

7.2.4 Dependent Metamodeling Research Thrust

In the proposed approach in Chapter 6, the correlation among responses from a multi-response deterministic simulation is used to build metamodels for all these responses simultaneously. Also, in the DOE stage, the experiments are selected based on the information from both the input space and the response space.

A numerical and an engineering example are used to show the strengths of the proposed dependent metamodeling approach. In these test problems, the multiple responses are correlated. And the correlation is used in building the metamodels for the responses simultaneously. It is shown from the two examples that, for the same number of experiments and in terms of the metamodel accuracy, the new approach outperforms a conventional method by a good margin.

One shortcoming of the proposed dependent metamodeling approach is that it's computational overhead (e.g., for solving the system of equations accounting for the correlation) is increased as the number of responses increase. Another shortcoming is that the DOE method used in the dependent metamodeling approach is not fully adaptive to each response and thus may limit the improvement of the metamodel accuracy.

7.3 Main Contributions

The main challenge in metamodeling assisted fitness evaluation is to determine, in each generation of the optimization process, the response from which points should be estimated using the metamodel. A kriging assisted fitness evaluation approach is developed in this dissertation in which the *MMD* and kriging error is used to identify such points. It is shown that the new approach is able to reduce the number of function calls significantly compared to a conventional population based optimization approach.

- *Kriging assisted fitness evaluation approach is the first online metamodeling approach that uses the uncertainty of the estimated responses as the determination criterion. Only for points with large estimated uncertainty in the response, the simulation is used to obtain the actual response. These evaluated points are used to update and improve the accuracy of the kriging metamodel gradually, thus more points can be processed by the metamodel in the later generations and thus save in simulation costs.*

In the Adaptive DOE for online metamodeling, there are two major challenges: 1) Selecting some points that can help the optimization process to converge faster, while keep the global searching ability of a population based optimization method, and 2) Being able to work in concert with a metamodel assisted fitness evaluation approach. A new DOE method is developed in this dissertation and is integrated in a conventional population based optimizer together with the fitness estimation approach presented in Chapter 4. It is shown that the savings obtained are significant in terms of both the number of function calls from both the DOE addition and the fitness estimation addition for a population based optimization framework.

- *The main contribution of this approach is that for the first time in the literature online metamodeling for both fitness estimation and reproduction has been integrated and used to significantly improve the performance for a population based optimization approach such as MOGA.*

In the Adaptive DOE for offline metamodeling, the major challenge is to select the minimum number of experiments to build accurate metamodel. This problem can be posed as an optimal selection problem that can be solved using a greedy algorithm. The results from the test examples in Chapter 5 show that the new approach outperforms a conventional DOE approach.

- *ACE approach is the first DOE approach in which the greedy principle is used explicitly to select new experiments in non-smooth regions which are identified using actual responses. In this way, the improvement of metamodel accuracy is maximized as more experiments are selected for the identified non-smooth regions.*

Finally in the metamodeling for deterministic simulations with multiple responses, as discussed in Chapter 6, the major challenge is to build accurate metamodels for all responses using as few as possible experiments. A novel dependent metamodeling approach is developed in this dissertation that uses the correlation among all responses from a simulation. It is shown that, for the same number of experiments, the new approach can build more accurate metamodels when compared to a conventional approach as demonstrated in Section 6.4.1 and Section 6.4.2.

- *The main contribution of this part of research is that the correlation among all responses is accounted in building metamodel for a multi-response simulation model. These metamodels are created assuming that other response values from the model can be used as inputs, thus creating a system of dependent metamodels that are solved simultaneously.*

7.4 Future Research Directions

In this section, several suggestions are given for future research. Some of these suggestions are based on the shortcomings discussed in the last section. Others are made based on extended research.

- **Kriging assisted fitness evaluation:** In Section 7.1.1, it was mentioned that the online metamodeling approach of Chapter 3 is conservative because the relation between *MMD* and kriging error was derived from a worst case scenario. Hence, the number of the function calls in each generation can be further reduced if a less conservative criterion can be developed. Also, the reduction in the number of function calls for a test problem is known only after the approach is applied to that problem. It would be useful if it is possible to establish a lower bound for the reduction so that the saving can be estimated before the start of the approach. Finally, although we observed that the K-MOGA approach can reduce the number of function calls for every test problem compared with the conventional MOGA approach, a theoretical proof is absent. The challenge is that the reduction of the number of function calls is problem dependent and this aspect needs to be further verified.

- **Adaptive DOE for online metamodeling:** In the DOE method of Chapter 4, the criterion for selecting some points from the predicted optima in the DOE method is similar to that used in the online metamodeling approach of Chapter 3. That is, the criterion is conservative and tends to select more points (with a large kriging error) from the predicted optima as DOE points for the next population. As discussed in Section 4.3.2, these DOE points are all evaluated by the simulation. As a result, the reduction in the number of function calls due to an online metamodeling assisted fitness evaluation can be diminished. Hence, a less conservative selection criterion can improve the efficiency of the proposed DOE method, too. Similarly, the determination of a lower bound for the reduction in the number of the generations for convergence will be helpful. The generality of the proposed approach in terms of reducing the number of function calls needs to be further verified.
- **ACE for DOE:** Another future research idea for adaptive DOE in offline metamodeling is the so called resource (or function call) allocation problem. That is, given a limited number of function calls, how many of these function calls should be allocated for initial design, and how many for metamodel validation (i.e., the metamodel accuracy, such as by the RMSE).
- **Dependent metamodeling:** As we mentioned in Section 7.1.4, the dependent metamodeling approach is not fully adaptive in the sense that the new experiments

are selected without using certain information from each response metamodel. Thus, the metamodel accuracy for a multi-response simulation can be further improved by an adaptive DOE method, such as the one presented in Chapter 5.

- **Approximation for simulations with uncertainty:** The approximation research presented in this dissertation assumes that the simulations are deterministic even though some uncertainty can exists in real simulations. The extension of the proposed approximation techniques in this dissertation to simulations with uncertainty would be an interesting research area. Some initial research on this aspect has been reported (e.g., Daniel et al., 2006).
- **Approximation for multiple coupled simulations:** Approximation methods developed in this dissertation are for all-at-once single simulations. However, an engineering system can be very complicated, have multiple disciplines, and involve multiple coupled simulations for those disciplines. Thus, a future challenge is how to efficiently extend the approximation methods in this dissertation to multi-subsystem and coupled simulations.

References

1. Abboud, K., Schoenauer, M., 2001, “Surrogate Deterministic Mutation: Preliminary Results,” *Lecture Notes in Computer Science*, Vol. 2310, pp. 104-116.
2. Alvarez-Benitez, J. E., Everson, R. M., Fieldsend, J. E., 2005, “A MOPSO Algorithm based Exclusively on Pareto Dominance Concepts,” *Evolutionary Multi-Criterion Optimization*, Vol. 3410, pp. 459-73.
3. Anderson, K. S., Hsu, Y., 1999, “Genetic Crossover Strategy Using an Approximation Concept,” in *Proceedings of IEEE Congress on Evolutionary Computation*, Washington DC, IEEE, pp. 527-533.
4. Arora, J. S., 2004, *Introduction to Optimum Design*, 2nd ed., Elsevier, New York.
5. Barthelemy, J.-F. M., Haftka, R. T., 1993, “Approximation Concepts for Optimum Structural Design-A Review,” *Structural Optimization*, Vol. 5, pp. 129-144.
6. Bazaraa, M. S., Sherali, H. D., Shetty, C. M., 1993, *Nonlinear Programming*: *Theory and Algorithms*, John Wiley & Sons, New York.
7. Bedewi, Z. A., Kan, A., Marzougui, D., 1996, “Validation of a Non-linear Finite Element Vehicle Model Using Multiple Impact Data,” *ASME Winter Annual Congress and Exposition*, Atlanta, GA.
8. Belegundu, A. D., Chandrupatla, T. R., 1999, *Optimization Concepts and Applications in Engineering*, Prentice Hall, New Jersey.
9. Chiao, C. H., Hamada, M. S., 2001, “Analysis of Experiments with Correlated Multiple Responses,” *Journal of Quality Technology*, Vol. 33, pp. 451-465.
10. Chung, H., Alonso, J. J., 2002, “Using Gradients to Construct Cokriging Approximation Models for High-Dimensional Design Optimization Problems,” in

Proceedings of the 40th AIAA Aerospace Sciences Meeting & Exhibit, Reno, NV.

11. Chung, H., Alonso, J., 2004, “Multiobjective Optimization Using Approximation Model-Based Genetic Algorithms,” in *Proceedings of the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, NY.
12. Clark, S. M., Griebsch, J. H., Simpson, T. W., 2005, “Analysis of Support Vector Regression for Approximation of Complex Engineering Analyses,” *Transactions of the ASME*, *Journal of Mechanical Design*, Vol. 127, pp. 1077-1087.
13. Coello Coello, A. C., Pulido, G. T., Lechuga, M. S., 2004. “Handling Multiple Objectives with Particle Swarm Optimization,” *IEEE Transactions on Evolutionary Computation*, Vol. 8, No. 3, pp. 256-279.
14. Coello Coello, C. A., 2000, “An Updated Survey of GA-based Multiobjective Optimization Techniques,” *ACM Computing Surveys*, Vol. 32, No. 2, pp.109-143.
15. Coit, D. W., Smith, A. E., Tate, D. M., 1996, “Adaptive Penalty Methods for Genetic Optimization of Constrained Combinatorial Problems,” *INFORMS Journal on Computing*, Vol. 8, pp. 173-182.
16. Cormen, T. H., Leiserson, C. E., Rivest R. L., Stein, C., 2001, *Introduction to Algorithms*, 2nd ed., the MIT Press, Cambridge, MA.
17. Cox, D. D., John S., 1992, “A Statistical Method for Global Optimization,” in *Proceedings of Systems, Man and Cybernetics, IEEE International Conference*, Chicago, IL.
18. Cox, D. D. John S., 1997, “SDO: A Statistical Method for Global Optimization,” *Multidisciplinary Design Optimization: State of the Ar*t, edited by N. Alexandrov and M. Y. Hussaini, SIAM, Philadelphia, PA, pp. 315-329.

19. Currin, C., Mitchell, M., Morris, M., Ylvisaker, D., 1988, “A Bayesian Approach to the Design and Analysis of Computer Experiments,” Oak Ridge National Laboratory, Rept. ORNL-6498.

20. Currin, C., Mitchell, M., Morris, M., Ylvisaker, D., 1991, “Bayesian Prediction of Deterministic Functions, with Applications to the Design and Analysis of Computer Experiments,” *Journal of the American Statistical Association*, Vol. 86, pp. 953-963.

21. Daniel, W. A., Liu, J., Chen, W., 2006, “Understanding the Effects of Model Uncertainty in Robust Design With Computer Experiments,” *Transactions of the ASME*, *Journal of Mechanical Design*, Vol. 128, pp. 945-958.

22. Deb, K., 1999, “Evolutionary Algorithms for Multi-criterion Optimization in Engineering Design,” In K. Miettinen, M. M. Makela, P. Neittaanmaki & J. Periaux (Eds.), *Evolutionary Algorithms in Engineering and Computer Science: Recent Advances in Genetic Algorithms, Evolution Strategies, Evolutionary Programming, Genetic Programming, and Industrial Applications*, John Wiley & Sons, New York.

23. Deb, K., 2001, *Multiobjective Optimization Using Evolutionary Algorithms*, John Wiley and Sons, New York.

24. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T., 2000, “Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II,” in *Proceedings of the 6th International Conference on Parallel Problem Solving from Nature* (*PPSN-VI*), Paris, France.

25. Del-Castillo, E. D., Montgomery, D. C., McCarthy, D. R., 1996, “Modified

Desirability Function for Multiple Response Optimization", *Journal of Quality Technology*, Vol. 28, No. 3, pp. 337-345.

26. Derringer, G., Suich, R., 1980, "Simultaneous Optimization of Several Response Variables", *Journal of Quality Technology*, Vol. 12, No. 4, pp. 214-219.
27. Douglas, J. M, 1988, *Conceptual Design of Chemical Processes*, MacGraw-Hill Chemical Engineering Series, NY.
28. Esquivel, S. C., Coello Coello, C. A., 2004, "Particle Swarm Optimization in Non-stationary Environments," *Lecture Notes in Computer Science*, Vol. 3315, pp. 757-766.
29. Fang, K.-T., Wang, Y., *Number-Theoretic Methods in Statistics*, Chapman & Hall, New York, 1994.
30. Fang, K.-T., Lin, D. K. J., Winker, P. Zhang, Y., 2000, "Uniform Design: Theory and Application," *Technometrics*, Vol. 42, pp. 237-248.
31. Fang, H., Rais-Rohani, M., Horstemeyer, M., 2004, "Multiobjective Crashworthiness Optimization with Radial Basis Functions," in *Proceedings of the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, NY.
32. Farhang-Mehr, A., Azarm, S., 2002, "A Sequential Information-Theoretic Approach to Design of Computer Experiments," in *Proceedings of the 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Atlanta, GA.
33. Farhang-Mehr, A., Azarm, S., Diaz, A., Ravisekar, A., 2003, "Approximation-Assisted Crashworthiness Design of Front-End of a Pickup Truck," in *Proceedings*

of the ASME 2003 Design Engineering Technical Conference, Chicago, IL.

34. Farhang-Mehr, A., Azarm, S., 2005, "Bayesian Metamodeling of Engineering Design Simulations: A Sequential Approach with Adaptation to Irregularities in the Response Behavior," *International Journal for Numerical Methods in Engineering,* Vol. 62, pp. 2104-2126.
35. Farina, M., 2001, "A Minimal Cost Hybrid Strategy for Pareto Optimal Front Approximation," *Evolutionary Optimization*, Vol. 3, No. 1, pp. 41-52.
36. Farina, M., 2002, "A Neural Network Based Generalized Response Surface Multiobjective Evolutionary Algorithms," in *Proceedings of Congress on Evolutionary Computation, IEEE Press*, pp. 956-961.
37. Fonseca, C. M., Fleming, P. J., 1993, "Genetic Algorithms for Multi-objective Optimization: Formulation, Discussion, and Generalization," in *Proceedings of the 5th International Conference on Genetic Algorithms*, Urbana-Champaign, IL.
38. Goldberg D. E., 1989, *Genetic Algorithm in Search, Optimization and Machine Learning*, Addison-Wesley, MA.
39. Guanwan, S., 2004, *Parameter Sensitivity Measures for Single Objective, Multi-Objective, and Feasibility Robust Optimization*, Ph.D. Dissertation, University of Maryland, College Park, MD.
40. Holland, J. H., 1975, *Adaptation in Natural and Artificial Systems*, University of Michigan, Ann Arbor, MI.
41. Hong, Y-S, Lee, H., Tahk, M-J., 2003, "Acceleration of the Convergence Speed of Evolutionary Algorithms Using Multilayer Neural Networks," *Engineering Optimization*, Vol. 35, No. 1, pp. 91-102.

42. Horn, J., Nafpliotis, N., 1993, “Multiobjective Optimization using the Niched Pareto Genetic Algorithm,” Technical Report No: IlliGAl Report 93005, University of Illinois at Urbana-Champaign, Urbana, Illinois.

43. Jeankoplis, C. J., 1993, *Transport Processes and Unit Operations*, 3rd ed., PTR Prentice-Hall, Englewood Cliffs, NJ.

44. Jin, R., Chen, W., Sudjianto, A., 2002. “On Sequential Sampling for Global Metamodeling in Engineering Design”, in *Proceedings of ASME IDETC*, *Design Automation Conference*, Montreal, Canada.

45. Jin, Y., 2005, “A Comprehensive Survey of Fitness Approximation in Evolutionary Computation,” *Soft Computing*, Vol. 9, No. 1, pp. 3-12.

46. Jin, Y., Olhofer, M., Sendhoff, B., 2001, “Managing Approximate Models in Evolutionary Aerodynamic Design Optimization,” in *Proceedings of IEEE Congress on Evolutionary Computation*, Vol. 1, pp. 592-599.

47. Jin, Y., Olhofer, M., Sendhoff, B., 2002, “A Framework for Evolutionary Optimization with Approximate Fitness Functions,” *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 5, pp. 48-494.

48. Johnson, M. E., Moore, L. M., Ylvisaker, D., 1990, “Minimax and Maximin Distance Designs,” *Journal of Statistical Planning and Inference*, Vol. 26, No. 2, pp. 131-148.

49. Jones, D. R., 2001, “A Taxonomy of Global Optimization Methods Based on Response Surfaces,” *Journal of Global Optimization*, Vol. 21, pp. 345-383.

50. Jones, D.R., Schonlau, M., Welch, W. J., 1998, “Efficient Global Optimization of Expensive Black Box Functions,” *Journal of Global Optimization*, Vol. 13, pp.

455-492.

51. Kennedy, J., Eberhart, R., 1995, "Particle Swarm Optimization," in *Proceedings of the 4th IEEE International Conference on Neural Networks*, Perth, Australia.

52. Khuri, A. I., 1996, *Multiresponse Surface Methodology, Handbook of Statistics*, Vol. 13, Elsevier, Amsterdam.

53. Kozen, D. C., 1992. *The Design and Analysis of Algorithms*, Springer-Verlag, New York, NY.

54. Kohavi, R., 1995, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection", in *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, Montréal, Québec, Canada.

55. Koehler, J. R., Owen, A. B., 1996, "Computer Experiments," *Handbook of Statistics*, Vol. 13, Elsevier Science, New York.

56. Koch, P. N., Wujek, B. A., Golovidov, O., Simpson, T.W., 2002, "Facilitating Probabilistic Multidisciplinary Design Optimization Using kriging Approximation Models," in *Proceedings of the 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Atlanta, Georgia.

57. Kurpati, A., Azarm, S., Wu, J., 2002, "Constraint Handling Improvements for Multi-Objective Genetic Algorithms," *Structural and Multidisciplinary Optimization*, Vol. 23, No. 3, pp. 204-213.

58. Li, G., Azarm, S., 2006, "Maximum Accumulative Error Sampling Strategy for Approximation of Deterministic Engineering Simulations", in *Proceedings of the 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Portsmouth, VA.

59. Li, G., Azarm, S., Farhang Mehr, A., Diaz, A., 2006a, "Approximation of Multiresponse Engineering Simulations: A Dependent Metamodeling Approach," *Structural and Multidisciplinary Optimization*, Vol. 31, No. 4, pp. 260-269.

60. Li, M., Li, G., Azarm, S., 2006b, "A Kriging Metamodel Assisted Multi-Objective Genetic Algorithm for Design Optimization", in *Proceedings of the ASME International Design Engineering Technical Conferences*, *the 32nd Design Automation Conference*, Philadelphia, PA.

61. Li, G., Li, M., Azarm, S., Rambo, J., Joshi, Y., 2007, "Optimizing Thermal Design of Data Center Cabinets with a New Multi-Objective Genetic Algorithm," *Distributed and Parallel Databases*, Vol. 21, No. 2-3, pp. 167-192.

62. Li, X., 2003, "A Non-dominated Sorting Particle Swarm Optimizer for Multiobjective Optimization," *Lecture Notes in Computer Science*, Vol. 2723, pp. 37-48.

63. Lian, Y., Liou, M., 2004, "Multiobjective Optimization Using Coupled Response Surface Model and Evolutionary Algorithm," in *Proceedings of the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, NY.

64. Lindley, D. V., 1956, "On a Measure of the Information Provided by an Experiment," *Ann Math Stat*, Vol. 27, pp. 986-1005.

65. Lophaven, S. N., Nielsen, H. B., Søndergaard, J., 2002, "Aspects of the Matlab Toolbox DACE," IMM-TR2002-13, Technical University of Denmark, DK.

66. Lucas, J. M., 1994, "How to Achieve a Robust Process Using Response Surface Methodology", *Journal of Quality Technology*, Vol. 26, No. 4 pp. 248-260.

67. Martin, J. D., Simpson, T. W., 2003, "A Study on the Use of Kriging Models to Approximate Deterministic Computer Models," in *Proceedings of the ASME IDETC*, *Design Automation Conference*, Chicago, IL.

68. McKay, M. D., Beckman, R. J., Conover, W. J., 1979, "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Response from a Computer Code," *Technometrics*, Vol. 21, No. 2, pp. 239-245.

69. Meckesheimer, M., Booker, A. J., 2002, "Computationally Inexpensive Metamodel Assessment Strategies," *AIAA Journal*, Vol. 40, No. 10, pp. 2053-2060.

70. Myers, R. H., and Montgomery, D. C., 1995, *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, John Wiley & Sons, New York.

71. Myers, R. H., Kim, Y., Griffiths, K., 1997, "Response Surface Methods with the Use of Noise Variables," *Journal of Quality Technology*, Vol. 29, pp. 429-440.

72. Nain, P., Deb, K., 2002, "A Computationally Effective Multiobjective Search and Optimization Technique Using Coarse-to-fine Grain Modeling," Indian Inst. Technol., Kanpur, Indian, Kangal Rep., 2002005.

73. Nain, P., Deb, K., 2003, "Computationally Effective Search and Optimization Procedure Using Coarse to Fine Approximations," in *Proceedings of the Congress on Evolutionary Computation (CEC-2003)*, Canberra, Australia, pp. 2081-2088.

74. Nair, P. B., Keane, A. J., 1998, "Combining Approximation Concepts with Genetic Algorithm-based Structural Optimization Procedures," in *Proceedings of the 39th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference and Exhibit, and AIAA/ASME/AHS Adaptive Structures Forum*, Long

Beach, CA.

75. Narayanan, S., Azarm, S., 1999, "On Improving Multiobjective Genetic Algorithms for Design Optimization," *Structural and Multidisciplinary Optimization*, Vol. 18, pp. 146-155.

76. Oduguwa, V., Roy, R., 2002, "Multiobjective Optimization of Rolling Rod Product design Using Metamodeling Approach," in *Proceedings of the Genetic Evolutionary Computation Conference*, pp. 1164-1171.

77. Owen, A. B., 1992, "Orthogonal Arrays for Computer Experiments, Integration and Visualization," *Statistica Sinica*, Vol. 2, pp. 439-452.

78. Pacheco, J. E., Amon, C. H., Finger, S., 2001, "Developing Bayesian Surrogates for Use in Preliminary Design", in *Proceedings of ASME Design Engineering Technical Conference*, *Design Theory and Methodology*, Pittsburgh, PA.

79. Papadrakakis, M., Lagaros, N., Tsompanakis, Y., 1999, "Optimization of Large-Scale 3D Trusses Using Evolution Strategies and Neural Networks," *International Journal of Space Structures*, Vol. 14, No. 3, pp. 211-223.

80. Papalambros, P. Y., Wilde, D. J., 2000, *Principles of Optimal Design*: *Modeling and Computation*, 2nd ed., Cambridge University Press, New York.

81. Rambo, J. Joshi, Y., 2005, "Reduced Order Modeling of Steady Turbulent Flows Using the Pod," in *Proceedings of HT2005 2005 ASME Summer Heat Transfer Conference*, San Francisco, CA.

82. Rasheed, K., 2000, "Informed Operators: Speeding Up Genetic-Algorithm-Based Design Optimization Using Reduced Models," in *Proceedings of Genetic and Evolutionary Computation Conference*, Las Vegas, pp. 628-635.

83. Rasheed, K., Ni, X., Vattam, S., 2005, "Comparison of Methods for Developing Dynamic Reduced Models for Design Optimization," *Soft Computing*, Vol. 9, pp. 29-37.

84. Reddy, M. J., Kumar, D. N., 2007, "An Efficient Multi-objective Optimization Algorithm based on Swarm Intelligence for Engineering Design," *Engineering Optimization*, Vol. 39, No. 1, pp. 49–68.

85. Rolander, N., Rambo, J., Joshi, Y., Mistree, F., 2005, "Towards Sustainable Design of Data Centers: Addressing the Lifecycle Mismatch Problem," in *Proceedings of IPACK05 International Electronic Packaging Technical Conference and Exhibition*, San Francisco, CA.

86. Romero, D. A., Amon, C. H., Finger, S., 2006, "On Adaptive Sampling for Single and Multi-Response Bayesian Surrogate Models," in *Proceedings of the ASME International Design Engineering Technical Conferences*, *the 32nd Design Automation Conference*, Philadelphia, PA.

87. Rosenberg, R. S., 1967, *Simulation of Genetic Populations with Biochemical Properties*, Ph.D. Dissertation, University of Michigan, Ann Arbor, MI.

88. Roux, W. J., Stander, N., Haftka, R. T., 1998, "Response Surface Approximations for Structural Optimization", *International Journal for Numerical Methods in Engineering*, Vol. 42, pp. 517-534.

89. Ruzika S., Wiecek, M. M., 2003, "A Survey of Approximation Methods in Multiobjective Programming," *Journal of Optimization Theory and Applications*, Vol. 126, No. 3, pp. 473-501.

90. Sacks, J., Welch, W. J., Mitchell, T. J., Wynn, H. P., 1989, "Design and Analysis

of Computer Experiments," *Statistical Science*, Vol. 4, No. 4, pp. 409-435.

91. Sasena, M. J., 2002, "Adaptive Experimental Design Applied to an Ergonomics Testing Procedure," in *Proceedings of the ASME IDETC*, *Design Automation Conference*, Montreal, Canada.
92. Sasena, M. J., Papalambros, P. Y., Goovaerts, P., 2000, "Metamodeling Sampling Criteria in a Global Optimization Framework," in *Proceedings of the 8th AIAA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Long Beach, CA.
93. Schaffer, J. D., 1985, "Multiple Objective Optimization with Vector Evaluated Genetic Algorithms," in *Proceedings of the First International Conference on Genetic Algorithms*, Pittsburgh, PA.
94. Schonlau, M., Welch, W. J., Jones, D. R., 1997, "Global versus Local Search in Constrained Optimization of Computer Models," *New Developments and Applications in Experimental Design*, edited by N. Flournoy, W. F. Rosenberger, and W. K. Wong, Institute of Mathematical Statistics.
95. Seader, J. D., Henley, E. J, 2006, *Separation Process Principles*, 2nd ed., John Wiley & Sons, NJ.
96. Shan, S., Wang, G.-G., 2005, "An Efficient Pareto Set Identification Approach for Multiobjective Optimization on Black-Box Functions," *Transactions of the ASME, Journal of Mechanical Design*, Vol. 127, pp. 866-874.
97. Shannon C. E., 1948, "A Mathematical Theory of Communication," *Bell System Technical Journal*, Vol. 27, pp. 379-423 and 623-656.
98. Shewry, M. C., Wynn, H. P., 1987, "Maximum Entropy Sampling," *Journal of*

Applied Statistics, Vol. 14, pp. 165-170.

99. Simpson, T. W., Booker, A. J., Ghosh, D., Giunta, A. A., Koch P. N., Yang, R.-J.,2004, “Approximation Methods in Multidisciplinary Analysis and Optimization: a Panel Discussion,” *Structural and Multidisciplinary Optimization*, Vol. 27, No. 5, pp. 302-313.

100. Simpson, T. W., Peplinski, J., Koch, P. N., Allen, J. K., 2001, “Metamodels for Computer-based Engineering Design: Survey and Recommendations,” *Engineering with Computers*, Vol. 17, No. 2, pp. 129-150.

101. Srinivas, N., Deb, K., 1994, “Multiobjective Optimization using Nondominated Sorting in Genetic Algorithms,” *Evolutionary Computation*, Vol. 2, No. 3, pp. 221-248.

102. Tekin, E., Sabuncuoğlu, I, 2004, “Simulation Optimization: A Comprehensive Review on Theory and Applications,” *IIE Transactions*, Vol. 36, pp. 1067–1081.

103. US Army Corps of Engineers Internet Publishing Group, 1997, “Engineering and Design – Practical Aspects of Applying Geostatistics at Hazardous, Toxic, and Radioactive Waste Sites,” Technical Letter, ETL 1110-1-175.

104. Ver Hoef, J. M., Cressie, N., 1993, “Multivariable Spatial Prediction,” *Mathematical Geology*, Vol. 5, No. 2, pp 219-240.

105. Vining, G. G., Myers, R. H., 1990, “Combining Taguchi and Response Surface Philosophies: A Dual Response Approach,” *Journal of Quality Technology*, Vol. 22, pp 38-45.

106. Wang, G. G., Shan, S., 2007, “Review of Metamodeling Techniques in Support of Engineering Design Optimization,” *Transactions of the ASME*, *Journal of*

Mechanical Design, Vol. 129, No. 4, pp. 370-380.

107. Watson, A. G., Barnes, R. J., 1995, "Infill Sampling Criteria to Locate Extremes," *Mathematical Geology*, Vol. 27, No. 5, pp. 589-608.

108. Weiss, M. A., 1997, *Data Structures and Algorithm Analysis in C*, 2nd ed., Addison-Wesley.

109. Wilson, B., Cappelleri, D., Frecker, M., Simpson, T. W., 2001, "Efficient Pareto Frontier Exploration Using Surrogate Approximations," *Optimization and Engineering*, Vol. 2, pp. 31-50.

110. Zitzler, E., Thiele, L., 1998, "An Evolutionary Algorithm for Multi-objective Optimization: The Strength Pareto Approach," No. Technical Report 43, Zurich, Switzerland: Computer Engineering and Communication Networks Laboratory, Swiss Federal Institute of Technology.