IOWA STATE UNIVERSITY
Digital Repository

Graduate Theses and Dissertations | Graduate College

2016

Multi-objective optimization of transonic airfoils using variable-fidelity models, co-kriging surrogates, and design space reduction

Anand Amrit
Iowa State University

Follow this and additional works at: http://lib.dr.iastate.edu/etd

 Part of the Aerospace Engineering Commons

Recommended Citation
Amrit, Anand, "Multi-objective optimization of transonic airfoils using variable-fidelity models, co-kriging surrogates, and design space reduction" (2016). *Graduate Theses and Dissertations*. 15148.
http://lib.dr.iastate.edu/etd/15148

This Thesis is brought to you for free and open access by the Graduate College at Iowa State University Digital Repository. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

www.manaraa.com

Multi-objective optimization of transonic airfoils using variable-fidelity models, co-kriging surrogates, and design space reduction

by

Anand Amrit

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Aerospace Engineering

Program of Study Committee:

Leifur Leifsson, Major Professor

Christina Bloebaum

Jonathan Regele

Iowa State University

Ames, Iowa

2016

Copyright © Anand Amrit, 2016. All rights reserved.

www.manaraa.com

DEDICATION

I would like to dedicate this thesis to my father Ajaya Kumar Nayak and to my mother Narayani Nayak who gave me emotional and moral strength to overcome all hurdles during my research. I would like to thank my advisor Dr. Leifur Leifsson whose ideas and efforts to make my research successful was instrumental. At the end, I would also like to thank my POSC committee members Dr. Christina Bloebaum and Dr. Jonathan Regele.

www.manaraa.com

TABLE OF CONTENTS

www.manaraa.com

www.manaraa.com

www.manaraa.com

LIST OF TABLES

www.manaraa.com

LIST OF FIGURES

www.manaraa.com

www.manaraa.com

www.manaraa.com

ABSTRACT

Computationally efficient constrained multi-objective design optimization of transonic airfoils is considered. The proposed methodology focuses on fixed-lift design aimed at finding the best possible trade-offs between the conflicting objectives. The algorithm exploits the surrogate-based optimization principle, variable-fidelity computational fluid dynamics (CFD) models, as well as auxiliary approximation surrogates (here, using kriging). The kriging models constructed within a reduced design space. The optimization process has three major stages: (i) design space reduction which involves the identification of the extreme points of the Pareto front through single-objective optimization, (ii) construction of the kriging model and an initial Pareto front generation using multi-objective evolutionary algorithm, and (iii) Pareto front refinement using co-kriging models. For the sake of computational efficiency, stages (i) and (ii) are realized at the level of low-fidelity CFD models. The proposed algorithm is applied to the multi-objective optimization of a transonic airfoil at a Mach number of 0.734 and a fixed lift coefficient of 0.824. The shape is parameterized with eight B-spline control points. The fluid flow is taken to be inviscid. The high-fidelity model solves the compressible Euler equations. The low-fidelity model is the same as the high-fidelity one, but with a coarser description and is much faster to execute. With the proposed approach, the entire Pareto front of the drag coefficient and the pitching moment coefficient is obtained using 100 low-fidelity samples and 3 high-fidelity model samples. This cost is not only considerably lower (up to two orders of magnitude) than the cost of direct high-fidelity model optimization using metaheuristics without design space reduction, but, more importantly, renders multi-objective optimization of transonic airfoil shapes computationally tractable, even at the level of accurate CFD models.

www.manaraa.com

NOMENCLATURE

$\mathbf{A}$	Response correction matrix
$A_{baseline}$	Baseline cross sectional area $[m^2]$
A_c	Cross sectional area of low-fidelity model $[m^2]$
A_f	Cross sectional area of high-fidelity model $[m^2]$
A_{min}	Minimum cross sectional area $[m^2]$
a_l	Scalar terms of response correction matrix $\mathbf{A}$
a_d	Scalar terms of response correction matrix $\mathbf{A}$
$\mathbf{C}_d$	Drag coefficient matrix of low-fidelity model
$\mathbf{C}_l$	Lift coefficient matrix of low-fidelity model
C_d	Drag coefficient [-]
$C_{d.c}$	Drag coefficient of low-fidelity model [-]
$C_{d.f}$	Drag coefficient of high-fidelity model [-]
C_l	Lift coefficient [-]
$C_{l.c}$	Lift coefficient of low-fidelity model [-]
$C_{l.f}$	Lift coefficient of high-fidelity model [-]
C_m	Pitching moment coefficient [-]
$C_{m.c}$	Pitching moment coefficient of low-fidelity model [-]
$C_{m.f}$	Pitching moment coefficient of high-fidelity model [-]
C_p	Pressure coefficient [-]
$\mathbf{D}$	Response correction matrix
D	Drag [N]
d_l	Scalar terms of response correction matrix $\mathbf{A}$
d_d	Scalar terms of response correction matrix $\mathbf{A}$
$\mathbf{F}_d$	Drag coefficient matrix of high-fidelity model

www.manaraa.com

$\mathbf{F}_l$	Lift coefficient matrix of high-fidelity model
$g(\mathbf{x})$	Inequality constraints
$h(\mathbf{x})$	Equality constraints
H	Scalar valued objective function
$\mathbf{l}$	Design variable lower bound
L	Lift [N]
M_∞	Mach number [-]
$\mathbf{q}$	Additive response correction
$\mathbf{s}$	Surrogate model
$\mathbf{u}$	Design variable upper bound
x	Airfoil chord-wise location [m]
X	B-spline control polygon coordinates
$\mathbf{x}$	Design variable
$\mathbf{x}^*$	Optimized design variable
z	Airfoil thickness [m]
Z	B-spline control polygon coordinates

www.manaraa.com

CHAPTER 1. INTRODUCTION

1.1 Motivation and Challenges

The development of complex engineering systems requires us to deal with various conflicting objectives. For example, in the development of a cellphone we need to look into several objectives like cost, battery life, and aesthetics. Attaining the best values of all the objectives simultaneous may be an impossible task. In such cases, we want to know the optimal decisions that need to be taken in the presence of trade-offs between the competing objectives. Here, the task is to find a representative set of optimal solutions that satisfy the different objectives, the so-called Pareto-optimal set [2], which describes the trade-offs between these objectives. Pareto optimality, is a state of allocation of resources in which it is impossible to make any one individual better off without making at least one individual worse off [3].

We will explain the concept of Pareto optimality through an example. Let us assume that we are designing an aircraft where we want to obtain the trade-off between two objectives $\mathbf{F} = [F_1 \quad F_2]^T$, where, for example, F_1 is the cost and F_2 is the range. Let us assume that there are two design variables $\mathbf{x} = [x_1 \quad x_2]^T$, where, for example, x_1 is the wing span and x_2 is the wing thickness to chord ratio. Figure 1.1 shows the feasible design space and the corresponding objective space. Consider designs A, B, C, and D. Out of these four designs, Design A gives the best range but at the same time it is the most expensive one. Similarly, Design B has a shorter range and is also very expensive. Design C, however, has the same range as B, but is much cheaper. Design D is cheaper than the other three designs, but has the same range as B and C. Moreover, Design D lies on the Pareto-optimal front, and represents the best design for the given values of the two objectives. In other words, it is not possible to find a design than has a longer range than Design D without increasing the cost.

www.manaraa.com

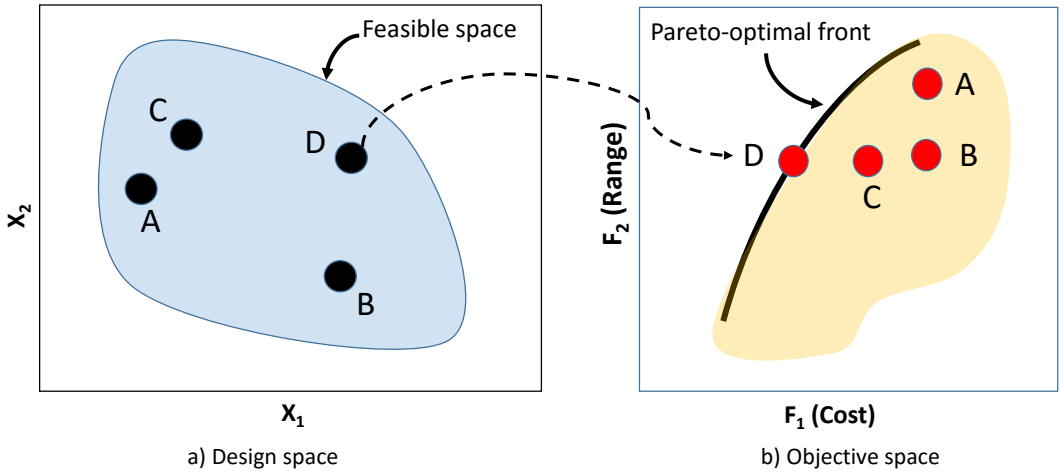

Figure 1.1 Representation of the design space and the corresponding feasible objective space. Designs A, B, and C are non-optimal solutions. Design D lies on the Pareto-optimal front.

The above example explains how the trade-offs between conflicting objectives can be represented within the design and objective spaces. A rudimentary approach to the simultaneous control of several objectives is a priori preference articulation (i.e., selection of the primary objective such as cost), and handling the remaining objectives by means of constraints of penalty functions. As a result, the problem can be solved as a single-objective one. However, in some situations it is of interest to gain more comprehensive information about the system at hand which may allow the designer to understand the characteristics of the possible trade-offs between conflicting objectives. In such a case, the entire Pareto front needs to be generated.

Multi-objective optimization [4] (MOO) (also called vector optimization or Pareto optimization) is used to obtain the trade-offs between competing objectives. There are various methods to perform MOO (Section 2 gives the background of MOO). A widely used approach involves the use of metaheuristic algorithms, such as genetic algorithms [5] (GAs), multi-objective evolutionary algorithm [6] (MOEAs), and particle swarm optimization [7] (PSO). Their primary advantage is the capability of generating the entire Pareto front representation in a single algorithm run. Unfortunately, metaheuristics are computationally intensive due to processing of large sets of candidate designs (population sizes of up to a few hundreds of individuals are not

www.manaraa.com

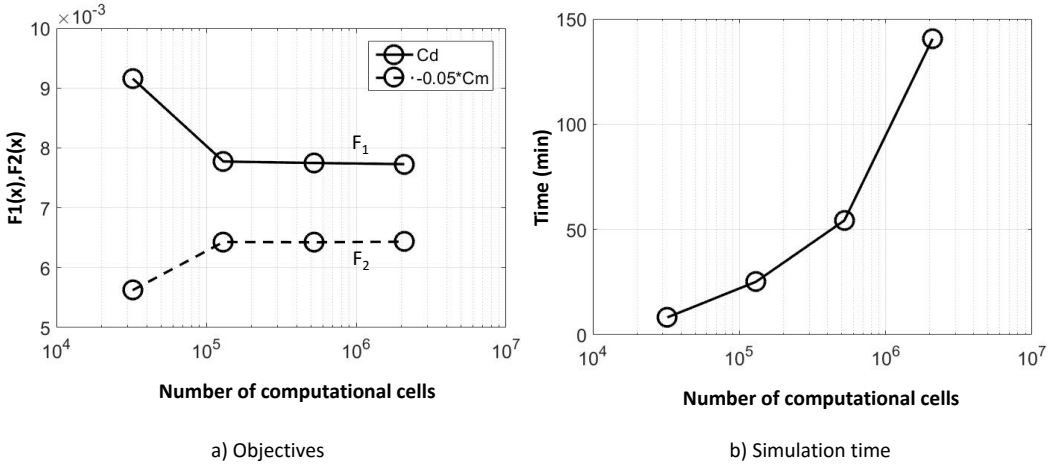

Figure 1.2 PDE simulations (such as computational fluid dynamics (CFD) simulations) need dense computational grids that require long CPU times (often on the order of days) on high performance computing clusters. The graph shows how the time and the objective function values of a 2D CFD simulation of transonic airfoil flow vary with the grid density.

uncommon). Consequently, metaheuristic algorithms are almost always limited to situations where the underlying computational model is very fast to execute and small design spaces.

High-fidelity partial differential equation (PDE) simulations are becoming increasingly important in the design of complex multidisciplinary engineering systems. The reason behind this is that the physics governing these complex systems can be highly nonlinear. Moreover, nonlinear couplings between disciplines may exist. Furthermore, when considering unconventional systems, it may not be possible to rely on prior designs. High-fidelity PDE simulations are, therefore, essential in the development of many modern engineering systems, even at the initial conceptual stage, since lower fidelity methods may not be able to capture reliably the main characteristics to yield the best design. Unfortunately, high-fidelity PDE simulations are computationally expensive. For example, a single PDE simulation of the fluid flow past an aerodynamic surface, such as a typical transonic transport wing shape, can be on the order of one day when using high performance computing (HPC). Figure 1.2 gives an example for the two-dimensional inviscid transonic flow past a airfoil. The trends for a wing shape will be the same, but the simulation time will be an order of magnitude larger.

www.manaraa.com

The key challenges with automated PDE-constrained MOO are as follows: (1) high computational cost of accurate PDE simulations, (2) large design space dimensionality and large parameter ranges, and (3) a large number of system evaluations required by conventional MOO techniques. The combination of (1) and (2) may yield design problems which are prohibitively expensive to solve using (3). Therefore, efficient methodologies are required to reduce the design space, speed-up the simulations while still retaining a desired accuracy, and reduce the number of required system evaluations.

1.2 Research Objectives and Contributions

The overall objective of this research work is to investigate strategies to accelerate PDE-constrained MOO to enable fast iterative system design. To limit the scope, we focus the work on the design of airfoil shapes in transonic fluid flow. This design problem requires the computational fluid dynamics (CFD) simulations of the transonic fluid flow. Airfoil shapes are typically described using up to 15 parameters, and, in this work we parameterize the shapes with eight parameters. The objective is to generate the trade-offs of the airfoil characteristics. In particular, the trade-offs between the drag and pitching moment coefficients are generated at a fixed lift coefficient. Therefore, the MOO design problem considered in this work involves small-scale PDE simulations (on the order of 20 minutes on HPC) and a low-dimensional design space (8 parameters). Future work, will consider larger scales and higher dimensional problems.

The MOO approach proposed in this work is as follows. We integrate fast physics-based surrogate models and design space reduction techniques to approximately identify the Pareto-optimal front, and, subsequently, refine the Pareto front using a limited number of computationally expensive system evaluations. To achieve this, we developed a computational framework which integrates variable-fidelity CFD models with numerical algorithms to perform the MOO. In particular, a critical step in the MOO process is to reduce the design space to enable the construction of an accurate approximation model (we use kriging interpolation [8]) with a limited number of fast low-fidelity CFD models. The design space reduction is achieved through single-objective optimization of each objective separately. The kriging surrogate is then utilized to generate the initial approximate Pareto front using MOEA, which is computationally

www.manaraa.com

efficient since the kriging surrogate is very fast. The approximate Pareto front is then refined by constructing a co-kriging model [8] on top of the initial kriging surrogate with a limited number of high-fidelity PDE simulations. To validate the approach, we perform MOO of the full design space and compare with the proposed approach.

1.3 Thesis Outline

The thesis is structured as follows. Chapter 2 provides the background of MOO, MOO algorithms, and MOO approaches used in aerodynamic design. The proposed MOO algorithm is described in Chapter 3. The results of the numerical application is given in Chapter 4. Chapter 5 concludes the thesis.

www.manaraa.com

CHAPTER 2. BACKGROUND

2.1 Definition and Formulation of Multi-Objective Optimization

The process of optimizing systematically and simultaneously a collection of objective functions is called multi-objective optimization (MOO) (also called vector optimization, Pareto optimization, or multi-criteria optimization) [2]. MOO is an area of multiple criteria decision making, that deals with mathematical optimization problems involving more than one objective function to be optimized simultaneously [4]. It has been applied in many fields of science, including engineering, economics and logistics where optimal decisions need to be taken in the presence of trade-offs between two or more conflicting objectives. Some of the few examples of multi-objective optimization problems involving two or three objectives are minimizing cost while maximizing comfort while buying a car, and maximizing performance whilst minimizing fuel consumption and emission of pollutants of a vehicle. Typically, the case of competitive objectives is the most interesting in the research field of MOP, because the choice of an "acceptable" or "best" solution relies on the trade-offs of the objective functions, which ultimately depends on human preferences and decisions [9].

MOO can be described in mathematical terms as follows [2, 3, 4]

$$\min(F_1(\mathbf{x}), F_2(\mathbf{x}), \ldots, F_k(\mathbf{x})) \tag{2.1}$$

$$\text{subject to } g_j(\mathbf{x}) \leq 0, j = 1, 2, \ldots, m, \tag{2.2}$$

$$h_l(\mathbf{x}) = 0, l = 1, 2, \ldots, e, \tag{2.3}$$

where k is the number of objective functions, m is the number of inequality constraints, and e is the number of equality constraints. $\mathbf{x} \in E^n$ is a vector of design variables (also called decision variables), where n is the number of independent variables x_i. $\mathbf{F}(\mathbf{x}) \in E^k$ is a vector

www.manaraa.com

of objective functions $F_i(\mathbf{x}) : E^n \rightarrow E^1$. $F_i(\mathbf{x})$ are also called objectives, criteria, payoff functions, cost functions, or value functions. The feasible design space X (often called the feasible decision space or constraint set) is defined as the set $\{\mathbf{x}|g_j(\mathbf{x}) \leq 0$, j=1 ,2,...,m; and $h_i(\mathbf{x}) = 0, i = 1, 2, ..., e\}$. The feasible criterion space Z (also called the feasible cost space or the attainable set) is defined as the set $\{\mathbf{F}(\mathbf{x})|\mathbf{x} \in X\}$.

Unlike single-objective optimization, a solution to a multi-objective problem is more of a concept than a definition. Typically, there is no single global solution, rather we may need to determine a set of points that all fit a predetermined definition for an optimum. The main concept in defining an optimal point is that of Pareto optimality [10]. All Pareto optimal points lie on the boundary of the feasible criterion space Z [11]. Often, algorithms provide solutions that may not be Pareto optimal but may satisfy other criteria, making them significant for practical applications. All Pareto optimal points may be categorized as being either proper or improper. The idea of proper Pareto optimality and its relevance to certain algorithms is discussed by [12, 13]. It is defined as follows: Properly Pareto Optimal: A point, $\mathbf{x} \in X$, is properly Pareto optimal (in the sense of [12]) if it is Pareto optimal and there is some real number $M \geq 0$ such that for each $F_i(\mathbf{x})$ and each $\mathbf{x} \in X$ satisfying $F_i(\mathbf{x}) < F_i(\mathbf{x}^\star)$, there exists at least one $F_j(\mathbf{x})$ such that $F_j(\mathbf{x}^\star) < F_j(\mathbf{x})$ and $(F_i(\mathbf{x}^\star) - F_i(\mathbf{x}))/(F_j(\mathbf{x}) - F_j(\mathbf{x}^\star)) \leq M$. The quotient is referred to as a trade-off, and it represents the increment in objective function j resulting from a decrement in objective function i. As required by the definition the trade-off between each function and at least one other function be bounded in order for a point to be properly Pareto optimal. For any given problem, there may be an infinite number of Pareto optimal points constituting the Pareto optimal set.

2.2 Multi-Objective Optimization Strategies and Algorithms

Several approaches and algorithms to solve MOO problems have been developed. The following is a brief description of several MOO approaches.

www.manaraa.com

2.2.1 Single-Objective Optimization using a Scalarized Objective Function

One way of solving the MOO problem is to scalarize the objectives and solve using single-objective optimization algorithms. The weighted sum method (WSM) is one way to scalarize the objectives. WSM transforms multiple objectives into an aggregated objective function by multiplying each objective function by a weighting factor and summing up all weighted objective functions as follows [14]

$$J_{weighted\ sum} = w_1 J_1 + w_2 J_2 + \cdots + w_m J_m, \tag{2.4}$$

where w_i, $i = 1, \ldots, m$, is a weighting factor for the ith objective function. The weight of an objective is chosen in proportion to the relative importance of the objective. The main disadvantage of this method is it is difficult to set the weight vectors to obtain a Pareto-optimal solution in a desired region in the objective space. Also it cannot find certain Pareto-optimal solutions in the case of a nonconvex objective space.

2.2.2 Evolutionary Algorithms

The term evolutionary algorithm (EA) stands for a class of stochastic optimization methods that simulate the process of natural evolution [6]. EAs have proven themselves as a general, robust and powerful search mechanism [15]. EAs seem to be especially suited to MOO because they are able to capture multiple Pareto-optimal solutions in a single simulation run and may exploit similarities of solutions by recombination. EAs operate on a set of candidate solution and use strong simplifications to subsequently modify the two basic principles of evolution: selection and variation. By 'selection' we mean the competition for resources among living beings. Some may be better than others and are more likely to survive and to reproduce their genetic information. In EAs, natural selection is simulated by a stochastic selection process. Each solution reproduces a certain number of time depending on their quality, which is assessed by evaluating the individuals and assigning them scalar fitness values. The other principle, 'variation', imitates natural capability of creating new living beings by means of recombination and mutation. According to some researchers multi-objective search and optimization might be a problem area where EAs do better than other blind search strategies [16]. Schaffer [17,

www.manaraa.com

18] performed the first studies on evolutionary MOO in the mid-1980s. This multi-objective evolutionary algorithm (MOEA) approach was later used in various field pertaining to multi-objective problems [19]. The main objectives of MOEAs are [6]: (a) the distance of the resulting non-dominated front to the Pareto-optimal front should be minimized, (b) a good (in most cases uniform) distribution of the solutions found is desirable, and (c) the spread of the obtained non-dominated front should be maximized, i.e., for each objective a wide range of values should be covered by the non dominated solutions.

2.2.3 Genetic Algorithms

Genetic algorithms (GAs) are one of the approaches that can be used to solve MOO problems directly. Holland [5] introduced GAs in 1975. Kocer [20] outlined a general GA and compared it with simulated annealing in its ability to minimize the cost of H-frame transmission poles subjected to earthquake loading with discrete variables. Gen and Cheng [21] used GAs to treat problems related to industrial engineering, whereas Davis [22] provided a more general treatment. Because GAs do not require gradient information, they can be effective regardless of the nature of the objective functions and constraints. They combine the use of random numbers and information from previous iterations to evaluate and improve a population of points (a group of potential solutions) rather than a single point at a time. GAs are global optimization techniques, which means they may converge to the global solution rather than to a local solution. However, it is not true when working with MOO, which usually entails a set of solution points. Mathematically, we cannot have a single global solution to a MOO problem. The GA methods involve global optimization, i.e, they determine solutions that are globally Pareto optimal, not just locally Pareto optimal. Schaffer [18] presents one of the first treatments of multi-objective GAs, although he only considers unconstrained problems. Schaffer's approach, which is also called the vector evaluated genetic algorithm (VEGA), involves producing smaller subsets of the original population, or sub-populations, within a given generation. Pareto optimality as a concept is not embedded in the fundamentals of GAs. It has no correlation to the natural origins of genetic methods and hence it is possible with certain multi-objective GAs that a Pareto optimal solution may be born and then die out.

www.manaraa.com

2.2.4 Particle Swarm Optimization

Particle Swarm Optimization (PSO) [7], another type of EA, simulates the movements of a flock of birds which aim to find food. The relative simplicity of PSO, and the fact that it is a population-based technique, have made it a natural candidate to be extended for MOO. Moore and Chapman proposed the first extension of the PSO strategy for solving multi-objective problems in an unpublished manuscript from 1999 [23]. There was great interest among researchers to extend PSO after this early attempt. The next proposal was not published until 2002. The main issues about extending PSO to MOO are discussed in [24]. In order to apply the PSO strategy for solving MOO problems, the original PSO scheme has to be modified. Reyes-Sierra and Coello [25] present a comprehensive review of the various multi-objective particle swarm optimization (MOPSO).

2.3 Applications of Multi-Objective Optimization in Aerodynamic Design

Shape optimization is an important part of the design of aerodynamic components such as aircraft wings and turbine blades [26, 27]. Nowadays, the use of high-fidelity computational fluid dynamic (CFD) simulations is widespread in aerodynamic design. While searching for an improved design using CFD-based parameter sweeps and engineering experience is still a common practice, design automation using numerical optimization techniques is becoming more and more popular [28, 29]. Various methods and algorithms are available, from conventional, gradient-based algorithms [30], including those utilizing cheap adjoint sensitivities [31, 32], to the more and more popular surrogate-based optimization techniques [33, 34, 35] that offer efficient global optimization, and substantial reduction of the design cost as compared to traditional methods [36].

Aerodynamic design is inherently a multi-objective problem. In many cases, a primary objective (e.g., drag coefficient minimization) may be selected through a priori preference articulation, whereas the others (e.g., lift coefficient) can be handled through design constraints. This allows for solving the problem as a single-objective one. Sometimes, however, this is neither possible nor convenient, e.g., when gaining knowledge about possible trade-offs between

www.manaraa.com

competing objectives (example drag coefficient and pitching moment coefficient) is important. In those situations, solving a genuine MOO is necessary.

The following are just a few examples from the literature where MOO is used in aerodynamic design. Shijun [37] uses the Davidon-Fletcher-Powell variable metric method as the multi-objective optimizer, and the Golden Section method for one-dimensional search, to maximize flutter speed by tailoring the fiber orientations of the skin and spar web laminates of an aircraft wing. Wesley [38] uses a GA with discrete design variable to construct an object-oriented multi-disciplinary aerodynamic optimization (MDAO) tool. The MDAO tool and high-fidelity structural analysis is used to minimize the structural weight while maintaining desired flutter speeds of an X-56A aircraft. Kai [39] presented a novel multidisciplinary framework for performing multi-objective shape optimization of a flexible wing structure. Using a multidisciplinary algorithm both aerodynamic shape and structural topology are optimized concurrently using gradient based optimization. Gaetan [40] uses the weighted sum method and a gradient-based search algorithm to perform an multi-objective aero-structural optimization. Ekhlas [41] uses multi-objective evolutionary algorithm (MOEA) for optimum aerodynamic design of horizontal-axis wind turbines (HAWT). Using the evolutionary method of combined GA and with different airfoil profiles technique, turbine aerodynamic performance is optimized. Hanafy [42] uses strength Pareto evolutionary algorithm (SPEA) based approach for designing an integrated fuzzy guidance law. Mukesha [43] uses PSO and GA to solve an aerodynamic shape optimization problem concerning NACA 0012 airfoil using 12 design variables. Results show that the PSO scheme is more effective in finding the optimum solution among the various possible solutions. Overall, based on this brief literature survey, it seems that for aircraft and aerodynamic design, the weighted sum method and GAs are generally used for MOO. Moreover, in these works the high-fidelity model is utilized directly in the MOO process.

www.manaraa.com

CHAPTER 3. MULTI-OBJECTIVE OPTIMIZATION METHODOLOGY

In this section, we define multi-objective aerodynamic design problem, and describe each step of the multi-objective optimization (MOO) algorithm proposed in this work. In particular, we describe the variable-fidelity computational fluid dynamics (CFD) modeling and space mapping correction, design space reduction technique, and the kriging and co-kriging surrogate model construction. The utilization of the MOO algorithm is demonstrated in Chapter 4 using an example of transonic airfoil shape design.

3.1 Multi-Objective Aerodynamic Design Formulation

We will denote by $\mathbf{x}$ the vector of design variables which are typically the geometry parameterization coefficients of the aerodynamic surface of interest. Also, let $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}) \quad f_2(\mathbf{x}) \dots f_k(\mathbf{x})]^T$ be a vector of k high-fidelity CFD model responses. Examples of responses include the airfoil section drag coefficient $f_1(\mathbf{x}) = C_{d.f}$, and the section lift coefficient $f_2(\mathbf{x}) = C_{l.f}$. Let $F_k(\mathbf{x})$, $k = 1, \dots, N_{obj}$, be the k^{th} design objective. A typical performance objective would be to minimize the drag coefficient ($C_{d.f}$), in which case $F_k(\mathbf{x}) = C_{d.f}$. Another objective would be to minimize the pitching moment coefficient ($C_{m.f}$), in which case $F_k(\mathbf{x}) = C_{m.f}$.

A comparison of the design solutions in a multi-objective setting is performed using a Pareto dominance relation. It is necessary because if $N_{obj} > 1$, then any two designs $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ for which $F_k(\mathbf{x}^{(1)}) < F_k(\mathbf{x}^{(2)})$ and $F_l(\mathbf{x}^{(2)}) < F_l(\mathbf{x}^{(1)})$ for at least one pair $k \neq l$, are not commensurable, i.e., none is better than the other in the multi-objective sense. We define Pareto dominance relation $\angle$ (see, e.g., Fonseca [2]), saying that for the two designs $\mathbf{x}$ and $\mathbf{y}$, we have $\mathbf{x} \angle \mathbf{y}$ ($\mathbf{x}$ dominates over $\mathbf{y}$) if $F_k(\mathbf{x}) \leq F_k(\mathbf{y})$ for all $k = 1, \dots, N_{obj}$, and $F_k(\mathbf{x}) < F_k(\mathbf{y})$ for at least one k. Multi-objective optimization aims at finding a representation of the Pareto

www.manaraa.com

front (of Pareto-optimal set) $\mathbf{X}_P$ of the design space $\mathbf{X}$, such that for any $\mathbf{x} \in \mathbf{X}_P$, there is no $\mathbf{y} \in \mathbf{X}$ for which $\mathbf{y} \measuredangle \mathbf{x}$ (Fonseca [2]). In practice, the Pareto front gives us information about the best possible trade-offs between the competing objectives, such as the minimum drag and pitching moment coefficients for a given value of the lift coefficient. Having a reasonable representation of the Pareto front is therefore indispensable in making various design decisions.

In practice, the identification of a set of alternative Pareto-optimal solutions needs to be followed by a decision making process, so that a single final design is selected for prototyping and subsequent manufacturing. This is done based on given preferences concerning, among others, importance of particular objectives. In this work, we only focus on the methodology for obtaining the Pareto front itself. The decision making process is beyond the scope of this work.

3.2 Optimization Algorithm

As mentioned in the introduction, a fundamental bottleneck of PDE-constrained optimization of complex systems is the high cost of the accurate, high-fidelity models, which is especially challenging in MOO. Therefore, for the sake of computational efficiency, the MOO procedure presented here exploits, apart from the original, high-fidelity model $\mathbf{f}$, its low-fidelity model counterpart $\mathbf{c}$. In this work, the low-fidelity model is based on coarse-discretization CFD simulations (its detailed setup is discussed in Chapter 4, which allows for a fast evaluation at the cost of some accuracy degradation. A design speedup is achieved by performing most of the operations at the level of the low-fidelity model; however, high-fidelity simulations are also executed in order to yield a Pareto set that is sufficiently accurate.

We present the entire MOO flow now, and then discuss each step in the following subsections. The proposed MOO algorithm is as follows:

1. This step is optional. Correct the low-fidelity model $\mathbf{c}$ using output space mapping to construct a surrogate model $\mathbf{s}_0$. If the correction is not performed, then $\mathbf{s}_0 = \mathbf{c}$;

2. Perform design space reduction using $\mathbf{s}_0$;

3. Sample the design space and acquire the surrogate model data with $\mathbf{s}_0$;

www.manaraa.com

4. Construct a kriging surrogate $\mathbf{s}_{KR}$ based on the data from Step 3;

5. Obtain an approximate Pareto set representation by optimizing $\mathbf{s}_{KR}$ using MOEA;

6. Evaluate the high-fidelity model $\mathbf{f}$ along the Pareto front;

7. Construct/update the co-kriging surrogate $\mathbf{s}_{CO}$;

8. Update Pareto set by optimizing $\mathbf{s}_{CO}$ using MOEA;

9. If termination condition is not satisfied go to Step 6; else END

Comments on each step in the above MOO algorithm:

Step 1: Searching for a Pareto front in a large design space using expensive high-fidelity PDE simulations is not practical, and, hence, a fast surrogate model will speed up the process. For the sake of reliability, output space mapping (Section 3.3) can be used to correct high-fidelity CFD models. However, this step may be skipped since the algorithm uses kriging (Step 4) and co-kriging (Step 7) to generate the Pareto front. These data-driven surrogates can provide the necessary alignment of the low-fidelity model with the high-fidelity one.

Step 2: Setting up an accurate data-driven surrogate (Step 4) can be very expensive to do in a large design space, i.e., wide parameter ranges, with multiple design variables. Hence, reducing the design space is a critical part of the propose MOO algorithm. With a smaller design space (in terms of reduced design variable parameter ranges, as well as reduced dimensionality), the kriging (Step 4) and co-kriging (Step 7) models can be set up accurately using few model evaluations. The design space reduction methodology is described in Section 3.4.

Step 3: Latin Hypercube Sampling (LHS) [44] is used to select the shapes within the reduced design space for the kriging model construction. The LHS sampling is described in Section 3.5.

Step 4: Using the sampled data from Step 3, a kriging interpolation model is constructed. Section 3.5 describes the surrogate model construction.

Step 5: A multi-objective evolutionary algorithm (MOEA) is run using the kriging model (from Step 4) to generate an initial approximation of the Pareto set. Note, however, the Pareto set obtained is not accurate to the high-fidelity level as we obtain it using the approximate

www.manaraa.com

surrogate which is based on the low-fidelity model. Here, we use a standard MOEA with fitness sharing, Pareto-dominance tournament selection, and mating restrictions [2]. The main changes (compared to single-objective evolutionary algorithms) are the mechanisms that push the solutions towards the Pareto front (here, realized through Pareto-dominance-based fitness function as well as the aforementioned selection procedure) and spread the solutions along the front (here, realized using fitness sharing and mating restrictions). The algorithm is modified in order to handle nonlinear constraints. The modification include: (i) initialization procedure that only generated feasible individuals, and (ii) crossover and mutation procedures that maintain feasibility of individuals.

Step 6 and 7: Designs are sampled uniformly along the Pareto front predicted by the initial kriging model optimization. Those designs are then evaluated using the high-fidelity model. A co-kriging model is then constructed (or updated) using all the high-fidelity model data accumulated during the algorithm run. Only a few high-fidelity model evaluations are used per iteration. The co-kriging model construction is discussed in Section 3.6.

Step 8: The co-kriging model is used to refine the Pareto front using the MOEA. In practice, just a few iterations are sufficient for convergence. If the alignment between these samples and the surrogate ones is sufficient, the algorithm is terminated. The convergence condition is based on the distance between the predicted front and the high-fidelity verification samples (distance measured in the feature space). In the case of the presented aerodynamic design problems, the threshold is set to 2 drag counts (one drag count is defined as $\Delta C_d = 0.0001$)).

3.3 Variable-Fidelity Surrogate Model

The main optimization engine utilized here for Pareto front identification is MOEA [45]. Due to excessive computational cost of population-based procedures such as MOEAs, it is not practical to apply evolutionary search directly at the level of the expensive PDE simulation model $\mathbf{f(x)} = [C_{d.f}(\mathbf{x})\ C_{m.f}(\mathbf{x})]^T$ (the lift coefficient $C_{l.f}(\mathbf{x})$ is kept constant by implicitly varying the angle of attack). Instead, we exploit a faster representation of the high-fidelity model, which is a surrogate constructed as follows. Let $\mathbf{c(x)} = [C_{d.c}(\mathbf{x})\ C_{m.c}(\mathbf{x})]^T$ denote the low-fidelity airfoil model, where $C_{d.c}$ and $C_{m.c}$ are the drag and pitching moment coefficients

www.manaraa.com

obtained by the low-fidelity CFD model (the low-fidelity lift coefficient $C_{l.c}(\mathbf{x})$ is kept constant by implicitly varying the angle of attack). The description and setup of the high- and low-fidelity models has been described in Chapter 4. The output space mapping (OSM) surrogate model $\mathbf{s}$ in Step 1 of the MOO algorithm (Section 3.2) is constructed as follows.

Enhancement of the low-fidelity model. In this step, the initial surrogate model $\mathbf{s}_0(\mathbf{x})$ is obtained by applying a parameterized output space mapping (OSM) [46, 47]. OSM uses correction terms that are directly applied to the response components $C_{d.c}(\mathbf{x})$ and $C_{m.c}(\mathbf{x})$ of the low-fidelity model (drag coefficient and pitching moment, respectively). The aerodynamic surrogate model is defined as [47]

$$\mathbf{s}_0(\mathbf{x}) = \mathbf{A}(\mathbf{x}) \circ \mathbf{c}(\mathbf{x}) + \mathbf{D}(\mathbf{x}), \tag{3.1}$$

where $\circ$ denotes component-wise multiplication, and $\mathbf{A}(\mathbf{x})$ and $\mathbf{D}(\mathbf{x})$ are multiplicative and additive correction terms. Both terms are design-variable-dependent and take the form of

$$\mathbf{A}(\mathbf{x}) = [a_{d.0} + [a_{d.1}a_{d.2}....a_{d.n}] \cdot (\mathbf{x} - \mathbf{x}^0) \quad a_{m.0} + [a_{m.1}a_{m.2}....a_{m.n}] \cdot (\mathbf{x} - \mathbf{x}^0)]^T, \tag{3.2}$$

$$\mathbf{D}(\mathbf{x}) = [d_{d.0} + [d_{d.1}d_{d.2}....d_{d.n}] \cdot (\mathbf{x} - \mathbf{x}^0) \quad d_{m.0} + [d_{m.1}d_{m.2}....d_{m.n}] \cdot (\mathbf{x} - \mathbf{x}^0)]^T, \tag{3.3}$$

where $\mathbf{x}^0$ is the center of the design space. Response correction parameters $\mathbf{A}(\mathbf{x})$ and $\mathbf{D}(\mathbf{x})$ are obtained as

$$[\mathbf{A},\mathbf{D}] = \arg\min_{[\bar{A},\bar{D}]} \sum_{k=1}^{N} || \mathbf{f}(\mathbf{x}^k) - (\bar{\mathbf{A}}(\mathbf{x}^k) \circ \mathbf{c}(\mathbf{x}^k) + \bar{\mathbf{D}}(\mathbf{x}^k)) ||^2 \tag{3.4}$$

i.e., the response scaling is supposed to (globally) improve the matching for all training points $\mathbf{x}^k$, $k = 1, \ldots, N$.

In this work, we use a training set consisting of (i) factorial design with $2n + 1$ training points (n being the number of design variables) allocated at the center of the design space $\mathbf{x}^0$ $= (\mathbf{l} + \mathbf{u})/2$ ($\mathbf{l}$ and $\mathbf{u}$ being the lower and upper bound for the design variables, respectively), and the centers of its faces, i.e., points with all coordinates but one equal to those of $\mathbf{x}^0$, and the remaining one equal to the corresponding component of $\mathbf{l}$ or $\mathbf{u}$; this sampling scheme is

www.manaraa.com

also referred to as the star distribution [36], (ii) additional 10 points allocated using the Latin Hypercube Sampling [44]. The only formal requirement for the necessary number of samples is that it is larger than the number of model coefficients to be identified.

The correction parameters $\mathbf{A}$ and $\mathbf{D}$ can be calculated analytically as follows [47]

$$\begin{bmatrix} a_{d.0} \\ a_{d.1} \\ \vdots \\ a_{d.n} \\ d_{d.0} \\ \vdots \\ d_{d.n} \end{bmatrix} = (\mathbf{C}_d^T\mathbf{C}_d)^{-1}\mathbf{C}_d^T\mathbf{F}_d \qquad \begin{bmatrix} a_{m.0} \\ a_{m.1} \\ \vdots \\ a_{m.n} \\ d_{m.0} \\ \vdots \\ d_{m.n} \end{bmatrix} = (\mathbf{C}_m^T\mathbf{C}_m)^{-1}\mathbf{C}_m^T\mathbf{F}_m \tag{3.5}$$

where

$$\mathbf{C}_d = \begin{bmatrix} C_{d.c}(\mathbf{x}^1) & C_{d.c}(\mathbf{x}^1)\cdot(x_1^1 - x_1^0) & \dots & C_{d.c}(\mathbf{x}^1).(x_n^1 - x_n^0) & 1 & (x_1^1 - x_1^0) & \dots & (x_n^1 - x_n^0) \\ C_{d.c}(\mathbf{x}^2) & C_{d.c}(\mathbf{x}^2).(x_1^2 - x_1^0) & \dots & C_{d.c}(\mathbf{x}^2)\cdot(x_n^2 - x_n^0) & 1 & (x_1^1 - x_1^0) & \dots & (x_n^1 - x_n^0) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ C_{d.c}(\mathbf{x}^N) & C_{d.c}(\mathbf{x}^N)\cdot(x_1^N - x_1^0) & \dots & C_{d.c}(\mathbf{x}^N)\cdot(x_n^N - x_n^0) & 1 & (x_1^1 - x_1^0) & \dots & (x_n^1 - x_n^0) \end{bmatrix} \tag{3.6}$$

$$\mathbf{F} = \begin{bmatrix} C_{m.f}(\mathbf{x}^1) & C_{m.f}(\mathbf{x}^2) & \dots & C_{m.f}(\mathbf{x}^N) \end{bmatrix}^T \tag{3.7}$$

which is a least-square optimal solution to the linear regression problems $[a_{d.0}\ a_{d.1}\ \dots a_{d.n}\ d_{d.0}\ d_{d.1}\ \dots\ \ d_{d.n}]^T\mathbf{C}_d = \mathbf{F}_d$ and $[a_{m.0}\ a_{m.1}\ \dots a_{m.n}\ d_{m.0}\ d_{m.1}\ \dots\ \ d_{m.n}]^T\mathbf{C}_m = \mathbf{F}_m$, equivalent to (3.4). Note that the matrices $\mathbf{C}_l^T\mathbf{C}_l$ and $\mathbf{C}_d^T\mathbf{C}_d$ are non-singular for $N > n+1$, which is the case for our choice of the training set.

The enhancement of the low-fidelity model using OSM can be skipped since it can be costly. The data-driven surrogates (Steps 4 and 7) may take care of the low-fidelity model alignment with the high-fidelity one. In this case, the surrogate model is simply

$$\mathbf{s}_0(\mathbf{x}) = \mathbf{c}(\mathbf{x}). \tag{3.8}$$

www.manaraa.com

3.4 Design Space Reduction

The MOO algorithm proposed in this work is heavily based on data-driven surrogate models. Therefore, it is of primary importance to ensure that the training data for creating the surrogate models can be acquired in a reasonable timeframe, even if the dimensionality of the design space is relatively large (say, more than 10 parameters). Here, similar as done in Koziel et al. [1], we carry out an initial design space reduction in order to identify the sub-region of the search space that contains the Pareto-optimal solutions. This region is usually a very small fraction of the original space. It is partially because the bounds for each geometry parameter are defined rather wide to ensure that the desired solutions are located within these prescribed limits. Nonetheless, setting up a data-driven surrogate in a large solution space may be impractical. However, the bounds of the design variables can be conveniently minimized using single-objective optimization runs with respect to each design goal. The result of those optimization runs should give us an approximation of where the extreme points of the Pareto-optimal set lie.

Consider $\mathbf{l}$ and $\mathbf{u}$ as the initial lower and upper bounds, respectively, for the design variable vector $\mathbf{x}$. Single-objective optimization of each objective F_k yield the approximate location of the extreme points $\mathbf{x}_c$ of the Pareto-optimal set, and can be found as

$$\mathbf{x}_c^{*(k)} = arg \min_{\mathbf{l} \leq \mathbf{x} \leq \mathbf{u}} F_k(\mathbf{s}_0(\mathbf{x})) \tag{3.9}$$

where $k = 1, \ldots, N_{obj}$. The boundaries of the reduced design space can be then defined as $\mathbf{l}^* = \min\{\mathbf{x}_c^{*(1)}, \ldots, \mathbf{x}_c^{*(N_{obj})}\}$ and $\mathbf{u}^* = \max\{\mathbf{x}_c^{*(1)}, \ldots, \mathbf{x}_c^{*(N_{obj})}\}$. The concept of the search space reduction is illustrated in Fig. 3.1. The reduced space is usually orders of magnitude (volume-wise) smaller than the initial one, which makes the generation of an accurate data-driven model possible at a reasonably low computational cost. Although some of the Pareto optimal solutions might fall outside the reduced design space, a majority of them are normally accounted for assuming that the objectives are continuous functions of the design variables and the Pareto front is a connected set.

www.manaraa.com

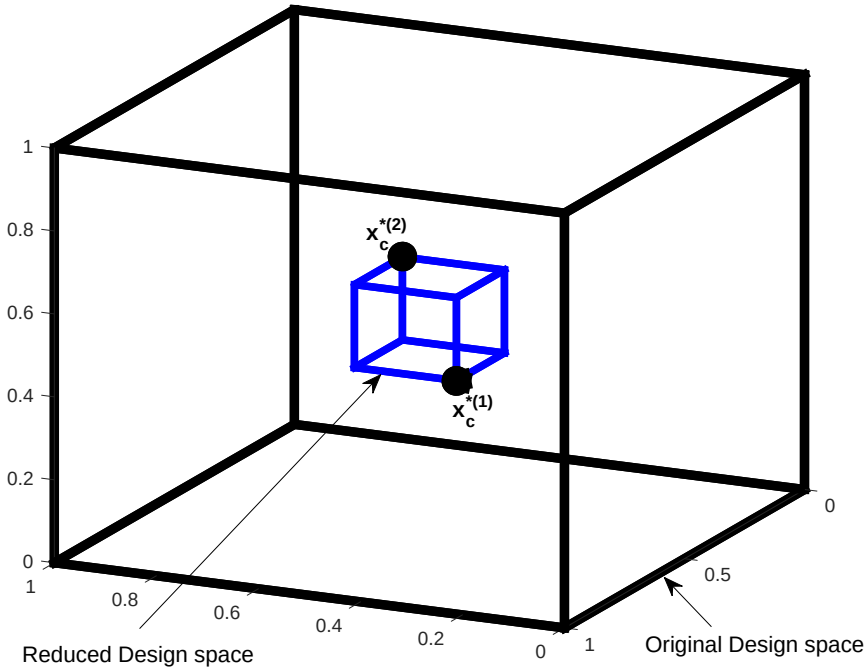

Figure 3.1 An illustration of the design space reduction technique using single-objective optimization runs (Koziel et al. [1]). The illustration assumes a three-dimensional design space and two design objectives.

3.5 Kriging Surrogate Construction

A kriging interpolation surrogate model is constructed in Step 4 of the MOO algorithm presented in Section 3.2. An outline of the process of constructing the surrogate is shown in Fig. 3.2 which involves performing design of experiments (DOE), data acquisition, model fitting, and model validation. Each step in the process is described in detail here below.

3.5.1 Design of Experiments

DOE is a technique to distribute sample points within the design space [35]. In this work, we use Latin Hypercube Sampling (LHS) [35]. LHS is a statistical method for generating a sample of a given parameter value from a multidimensional distribution, and ensures that each probability distribution in the model is evenly sampled. The idea of LHS is to use bins to sample the points along each design variable dimension. For example, as shown in Fig. 3.3, if

www.manaraa.com

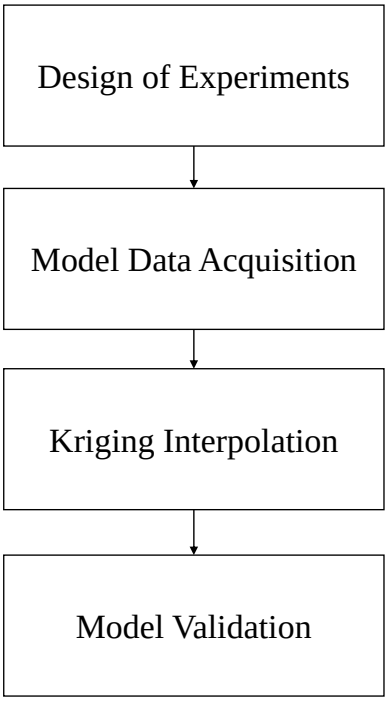

Figure 3.2 Flowchart describing the data-driven surrogate modelling process

the range of each design variable is split into 20 bins, for the two-dimensional case, there are 20^2 cells in the design space. The samples are allocated randomly so that for each dimension bin there is only one sample inside.

3.5.2 Kriging Interpolation

A kriging [8] interpolation model is utilized in this work to yield an initial approximation of the Pareto set. It is also the core of the co-kriging approach described in Section 3.6. This section briefly provides background information on kriging interpolation. A detailed survey can be found in the literature [48, 49].

Let $X_B = \{\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^N\}$ be a training set, and let $\mathbf{f}(X_{B.KR})$ be the corresponding set of high-fidelity model responses. The aim of kriging interpolation is to fit a regression function on the training nodes and build a Gaussian Process (GP) through the residuals [8]. The regression function $\mathbf{s}_{KR}(\mathbf{x})$ captures highest variance in the training samples while the GP covers the

www.manaraa.com

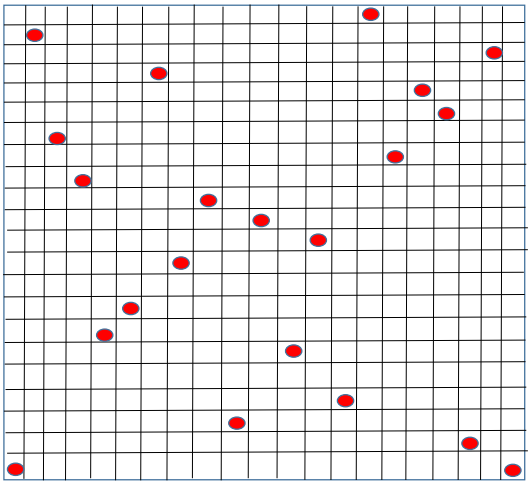

Figure 3.3 Latin Hypercube Sampling Illustration

details related to the interpolation accuracy. This is provided by a kriging interpolant denoted as

$$\mathbf{S}_{KR}(\mathbf{x}) = \mathbf{M}\alpha + \mathbf{r}(\mathbf{x}) \cdot \mathbf{\Psi}^{-1} \cdot (\mathbf{f}(X_{B.KR}) - \mathbf{F}\alpha), \tag{3.10}$$

where $\mathbf{M}$ and $\mathbf{F}$ are model matrices of the test node $\mathbf{x}$ and the base set $X_{B.KR}$, respectively. The vector α is a regression function coefficient of the form

$$\alpha = (X'_{B.KR}\mathbf{\Psi}^{-1}X_{B.KR})^{-1}X_{B.KR}\mathbf{\Psi}^{-1}(\mathbf{f}(X_{B.KR})) \tag{3.11}$$

while $\mathbf{r}(\mathbf{x}) = (\mathbf{\Psi}(\mathbf{x}, \mathbf{x}^1_{KR}), ..., \mathbf{\Psi}(\mathbf{x}, \mathbf{x}^{N_{KR}}_{KR}))$ is an $1 \times N_{KR}$ vector of correlations between the point $\mathbf{x}$ and the base set $X_{B.KR}$, and $\mathbf{\Psi}$ is a $N_{KR} \times N_{KR}$ correlation matrix given by

$$\begin{bmatrix} \mathbf{\Psi}(\mathbf{x}^1_{KR}, \mathbf{x}^1_{KR}) & \dots & \mathbf{\Psi}(\mathbf{x}^1_{KR}, \mathbf{x}^{N_{KR}}_{KR}) \\ \vdots & \ddots & \vdots \\ \mathbf{\Psi}(\mathbf{x}^{N_{KR}}_{KR}, \mathbf{x}^1_{KR}) & \dots & \mathbf{\Psi}(\mathbf{x}^{N_{KR}}_{KR}, \mathbf{x}^{N_{KR}}_{KR}) \end{bmatrix} \tag{3.12}$$

The kriging interpolation model is capable to predict the approximation error at any location in the solution space. The error is zero at the training nodes, which is because kriging is an interpolative model. The regression function actually operates as the mean of the GP, thus the predictions situated too far from existing training nodes (e.g., outside the sampled region)

www.manaraa.com

will revert to the average values. The nature of the response is usually unknown so that a constant regression function (referred to as ordinary kriging) is often utilized. One should note that in such cases, kriging is solely an interpolation method without the possibility of response extrapolation. The choice of the proper correlation function is important in order to create an accurate kriging surrogate. A widely used class of correlation functions dependent only on the distance between any two points (here **x** and **x'**) is defined by:

$$\Psi(\mathbf{x},\mathbf{x'}) = exp\left(\sum_{k=1,\ldots,n} -\theta_k |x^k - x^k|^p\right), \tag{3.13}$$

where the parameter p determines the prediction smoothness, while θ_k, $k = 1, \ldots, n$, denotes the influence sphere of a node on its neighbors in each dimension [50]. This is helpful for identification of relevant variables as it describes the linearity of the response. Usually, p is constant while the parameters θ_k are determined using Maximum Likelihood Estimation (MLE) [51], where the negative concentrated log-likelihood is minimized using

$$\ln(L) \approx -N_{KR}/2 \times \ln(\sigma^2) - 1/2 \ln(\mid \mathbf{\Psi} \mid), \tag{3.14}$$

and

$$\sigma^2 = (\mathbf{f}(X_{B.KR}) - \mathbf{F}\alpha)' \mathbf{\Psi}^{-1} (\mathbf{f}(X_{B.KR}) - \mathbf{F}\alpha)/N_{KR}. \tag{3.15}$$

Usually, $p = 2$ is selected (also known as the Gaussian correlation function), which is suitable for many problems. In the case of sharp responses, selecting $p = 1$ (which corresponds to the exponential correlation function) is normally more suitable. Finally, because no extrapolation capabilities are required, the regression function is set to be constant, i.e., $\mathbf{F} = [1 \quad 1 \quad \ldots 1]^T$ and $\mathbf{M} = (\mathbf{1})$.

3.5.3 Model Validation

Model validation is needed to check the quality of the surrogate model constructed in the data fitting process. Apart from the sampled set of designs that are used to construct the kriging model, a randomly selected subset is set aside for model validation purposes. The main purpose of these sets is to allow us evaluate the difference between the true model and the

www.manaraa.com

kriging model values at the specified test sites. In this work, we use the root mean square error (RMSE) metric [35]. We take a set of data of the size n_t and a set of predictions at those locations and calculate the RMSE as

$$RMSE = \sqrt{\frac{\sum_{i=0}^{n_t}(y^{(i)} - \hat{y}^{(i)})^2}{n_t}} \tag{3.16}$$

where n_t is the size of test data, $y^{(i)}$ is the true data and $\hat{y}^{(i)}$ is the predicted function value by the surrogate. Generally, we want the RMSE metric to be as small as possible. A kriging model with a RMSE value less than 2% can be considered as a reasonably good model, and can be used reliably within a surrogate-based optimization process [35].

3.6 Co-kriging Surrogate Construction

In this work, combining variable-fidelity CFD simulation data into a single surrogate model is of primary importance for reducing the cost of the Pareto-optimal set identification. For that purpose, we follow the work of Koziel et al. [52] and use co-kriging [8]. Co-kriging is an extension of kriging which exploits correlations between the models of various fidelities. This results in a considerable enhancement of the surrogate prediction accuracy even if the number of high-fidelity data samples is very small compared to what is normally required by single-level (in particular, conventional kriging) modeling. Here, the autoregressive co-kriging model of Kennedy et al. [53] is adopted.

The generation of the co-kriging model is carried out through a sequential construction of two kriging models: the first model $\mathbf{s}_{KRc}$ dervied from the original surrogate training samples $(X_{B.KRc}, \mathbf{c}(X_{B.KRc}))$, and the second $\mathbf{s}_{KRd}$ model generated on the residuals of the high- and low-fidelity samples $(X_{B.KRf}, s_d)$, where $\mathbf{s}_d = \mathbf{f}(X_{B.KRf}) - \rho\mathbf{c}(X_{B.KRf})$. The parameter ρ is a part of MLE of the second model. If $\mathbf{c}(X_{B.KRf})$ is not available, it can be approximated by the first model, i.e., as $\mathbf{c}(X_{B.KRf}) \approx \mathbf{s}_{KRc}(X_{B.KRf})$. One should emphasize that the configuration (specifically, the choice of the correlation function, regression function, and so forth) of both models can be adjusted separately for the low-fidelity data $\mathbf{c}$ and the residuals $\mathbf{s}_d$, respectively. Both models use (3.13) as a correlation function together with constant regression function

www.manaraa.com

$\mathbf{F} = [1 \quad 1 \quad \dots \quad 1]^T$ and $\mathbf{M} = (1)$. The final co-kriging model $\mathbf{s}_{CO}(x)$ is defined similarly as in (3.10), i.e.,

$$\mathbf{s}_{CO}(\mathbf{x}) = \mathbf{M}\alpha + \mathbf{r}(\mathbf{x}) \cdot \mathbf{\Psi}^{-1} \cdot (\mathbf{S}_d - \mathbf{F}\alpha), \tag{3.17}$$

where the block matrices $\mathbf{M}$, $\mathbf{F}$, $\mathbf{r}(\mathbf{x})$, and $\mathbf{\Psi}$ of (3.14) can be written as a function of the two underlying kriging models $\mathbf{s}_{KRc}$ and $\mathbf{s}_{KRd}$ as

$$\mathbf{r}(\mathbf{x}) = [\rho \cdot \sigma_c^2 \cdot \mathbf{r}_c(\mathbf{x}) \quad \rho^2 \cdot \sigma_c^2 \cdot \mathbf{r}_c(\mathbf{x}, X_{B.KR_f}) + \sigma_d^2 \cdot \mathbf{r}_d(\mathbf{x})],$$

$$\Psi = \begin{bmatrix} \sigma_c^2 \mathbf{\Psi}_c & \rho \cdot \sigma_c^2 \cdot \mathbf{\Psi}_c(X_{B.KR_c}, X_{B.KR_f}) \\ 0 & \rho^2 \cdot \sigma_c^2 \cdot \mathbf{\Psi}_c(X_{B.KR_f}, X_{B.KR_f}) + \sigma_d^2 \cdot \mathbf{\Psi}_d \end{bmatrix},$$

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_c & 0 \\ \rho \cdot \mathbf{F}_d & \mathbf{F}_d \end{bmatrix},$$

and

$$\mathbf{M} = [\rho \cdot \mathbf{M}_c \;\; \mathbf{M}_d].$$

www.manaraa.com

CHAPTER 4. NUMERICAL APPLICATIONS

In this chapter, we demonstrate the multi-objective optimization (MOO) algorithm of Section 3.2 on the design of aerodynamic surfaces. The case considers transonic airfoil shapes with eight design variables, and two conflicting objectives. In particular, we generate the trade-offs for the drag and pitching moment coefficients at a constant lift coefficient. The chapter is organized as follows. The problem is described in detail first. Three variations of the proposed MOO algorithm (which we call Strategies 1, 2, and 3) are applied to the design problem and the results presented. The chapter ends with a comparison of the strategies, and a parametric study of the third strategy.

4.1 Problem Description

Here, we give the details of the following: formulation of the problem, the design variable parameterization, training data sampling, variable-fidelity computational fluid dynamics (CFD) modeling, and an outline of the investigations with the three strategies.

4.1.1 Formulation of the MOO Problem

We consider multi-objective airfoil design in transonic flow at fixed lift. In particular, the free-stream Mach number is set to $M_\infty = 0.734$, and the lift coefficient is fixed at $C_{l.f}(\mathbf{x}) = 0.824$, where the subscript f refers to the high-fidelity model, and $\mathbf{x}$ is the design variable vector with a lower bound $\mathbf{l}$ and an upper bound $\mathbf{u}$. The first objective is $F_1(\mathbf{x}) = C_{d.f}(\mathbf{x})$, where $C_{d.f}$ is the high-fidelity drag coefficient. The second objective is $F_2(\mathbf{x}) = C_{m.f}(\mathbf{x})$, where $C_{m.f}$ is the high-fidelity pitching moment coefficient. Both objectives are minimized. We impose a constraint on the cross-sectional area, i.e., we have $A(\mathbf{x}) \geq A_{baseline}$, where $A(\mathbf{x})$ the cross-

www.manaraa.com

sectional area of a given design $\mathbf{x}$ nondimensionalized with the chord squared, and $A_{baseline}$ is a baseline reference value.

4.1.2 Design Space

We use the B-spline parameterization approach to describe the shape of the airfoil. The design variable vector is $\mathbf{x} = \mathbf{p}$, where $\mathbf{p}$ is a vector of the size $m \times 1$, with m being the total number of control parameters. The airfoil surfaces are written in parametric form as [54]

$$x(t) = \sum_{i=1}^{n+1} X_i N_{i,k}(t), \quad z(t) = \sum_{i=1}^{n+1} Z_i N_{i,k}(t), \tag{4.1}$$

where (x, z) are the Cartesian coordinates of the airfoil surface, $N_{i,k}$ is the B-spline basis function of order k, (X_i, Z_i) are the coordinates of the B-spline control polygon, and $m = n+1$ is the total number of control points. Note that the surface description with (4.1) is continuous. The control points are used as design variables and allowed only to move freely vertically as shown in Fig. 4.1. Thus, we have $\mathbf{x} = [Z_1 \quad Z_2 \quad \dots \quad Z_{n+1}]^T$ and the corresponding X_i coordinates are fixed during the optimization process. In this work, we use 8 design variables with 4 for each surface (as shown in Fig. 4.1).

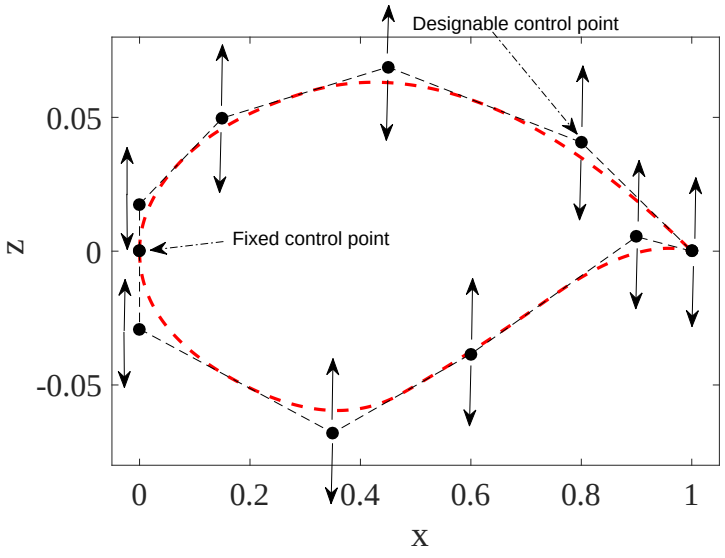

Figure 4.1 Example B-spline parameterization of an airfoil. The designable control points are restricted to vertical movements only.

www.manaraa.com

We use the RAE 2822 airfoil as a baseline. The shape is fitted to a B-spline curve by setting the x-locations of design variables as $\mathbf{X} = [\mathbf{X}_u;\ \mathbf{X}_l]^T$ = [0.0 0.15 0.45 0.8; 0.0 0.35 0.6 0.9]^T. After the fit, the baseline design variable vector is $\mathbf{x}_0 = [\mathbf{x}_u;\ \mathbf{x}_l]^T$ = [0.0175 0.04975 0.0688 0.0406; -0.0291 -0.0679 -0.03842 0.0054]^T. The bounds of the design space are defined by $\mathbf{l} = (1 - sign(\mathbf{x}_0) \cdot 0.15) \circ \mathbf{x}_0$ and $\mathbf{u} = (1 + sign(\mathbf{x}_0) \cdot 0.15) \circ \mathbf{x}_0$. Using the definition, the lower and upper bounds on $\mathbf{x}_0$ are set as $\mathbf{l}$ = [0.0105 0.0414 0.0537 0.0200; -0.0369 -0.0808 -0.0666 -0.0265]^T and $\mathbf{u}$ = [0.0231 0.0629 0.0889 0.0816; -0.0231 -0.0536 -0.0210 0.0140]^T, respectively. The baseline reference cross-sectional area is $A_{RAE2822}$= 0.0779.

4.1.3 Training Points

Sets of design points are generated around the baseline airfoil (the RAE 2822), within the upper and lower bounds, using Latin Hypercube Sampling (LHS) (see Section 3.5). Apart from LHS sampling, 256 corner points of the design space are generated and included in the set. Once the points are generated, each of them are checked for violation of area constraint and those infeasible are removed. An initial base set of 1,600 designs is used in all the strategies (this number is based on the mean square error and the values for each strategy are given in the results). Figure 4.2 shows the baseline airfoil, as well as a few samples from the base set. Figure 4.3 shows the design points (in black) for 2 of the control points.

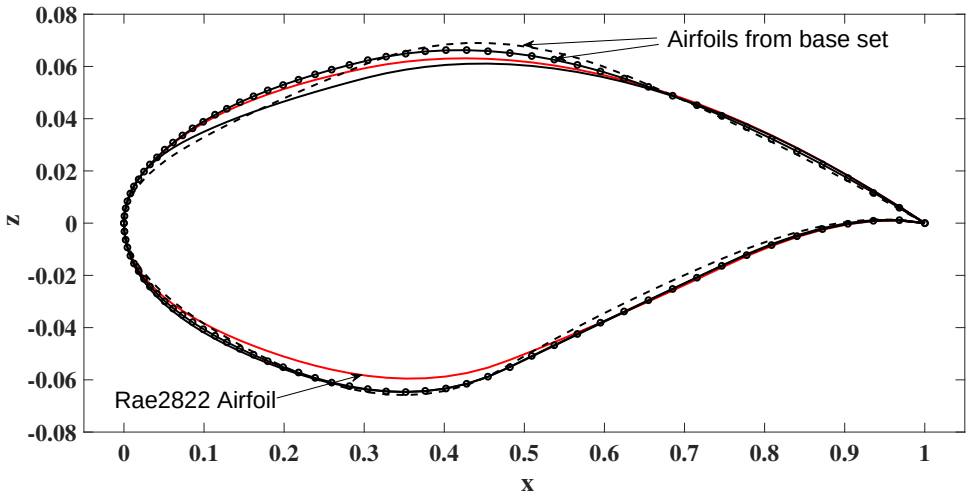

Figure 4.2 The baseline airfoil (RAE 2822) and sample airfoils from the base training set.

www.manaraa.com

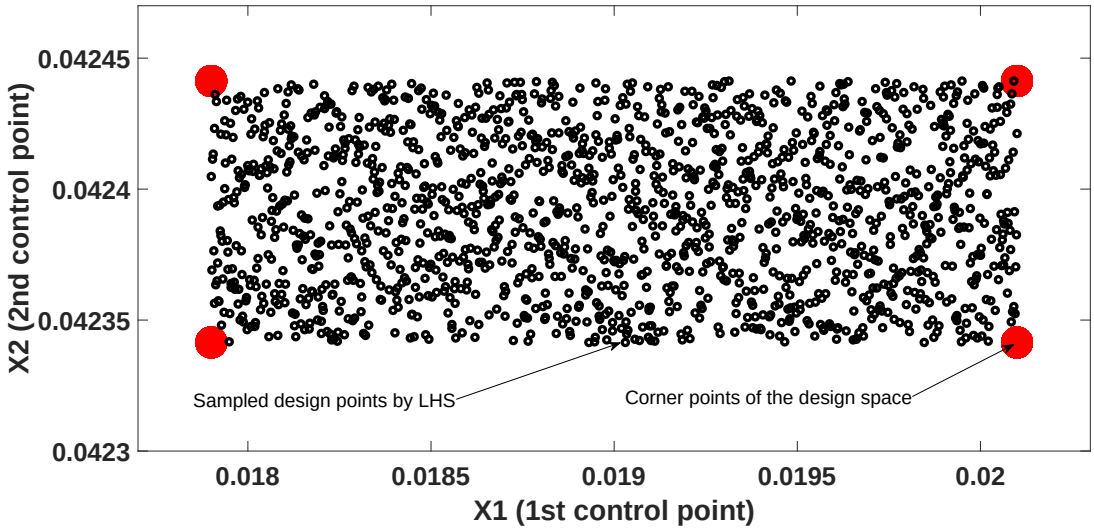

Figure 4.3 Example training points sampled using Latin Hypercube Sampling.

Output space mapping is utilized in Strategy 2 to setup the inital surrogate. In that case, a set of 30 points for the low- and high-fidelity simulations are generated within the given lower and upper bounds. Out of the 30 points, $2n + 1$ points are generated using a star distribution (these are at the centers of bounds and at the center, hence the name), and the rest are generated using LHS.

4.1.4 Computational Fluid Dynamics Modeling

The Stanford University Unstructured (SU2) computer code [55] is utilized for the fluid flow simulations. The steady compressible Euler equations are solved with an implicit density-based formulation. The convective fluxes are calculated using the second order Jameson-Schmidt-Turkel (JST) scheme [56]. Three multi-grid levels are used for solution acceleration. Asymptotic convergence to a steady state solution is obtained in each case. The flow solver convergence criterion is the one that occurs first of the two: (i) the change in the drag coefficient value over the last 100 iterations is less than 10^{-4}, or (ii) a maximum number of iterations of 1,000.

An O-type computational mesh is generated using Pointwise. The farfield boundary is set 55 chord lengths away from the airfoil surface. The mesh density is controlled by the number of

www.manaraa.com

cells on the airfoil surface and the number of cells normal to the surface. Distance to the first grid point is $0.001c$. The results of a grid convergence study, given in Table 4.1, revealed that the 512×512 mesh (shown number 5 in the table) is required for convergence within 0.2 drag counts (1 drag count is $\Delta C_d = 10^{-4}$) when compared with the next mesh. The flow simulation for Mesh 5 takes about 30 minutes. This time includes several simulations to obtain the desired lift coefficient by varying the angle of attack. Typically, 3 to 4 simulations are required.

For the multi-objective optimization studies, Mesh 5 will be used as the high-fidelity model $\mathbf{f}$, and Mesh 3 as the low-fidelity model $\mathbf{c}$. Figures 4.4, 4.5, and 4.6 show the computational grids. For the low-fidelity model, the maximum number of solver iterations is set to 300. Figure 4.1.5 shows the solver convergence of the low-fidelity model. Consequently, the high-to-low simulation time ratio is around 30 (see Fig. 4.9). A comparison of the pressure distributions, shown in Fig. 4.8, indicates that the low-fidelity model, in spite of being based on much coarser mesh and reduced flow solver iterations, captures the main features of the high-fidelity model pressure distribution quite well. The comparison indicates that the low-fidelity model may be a relatively good representation of the high-fidelity one. The biggest discrepancy in the distributions is around the shock on the upper surface, leading to an under estimation of both the drag and pitching moment coefficients (Table 4.1). Note that the drag and pitching moment coefficients are presented in terms of counts. We define one drag count (d.c.) to be $\Delta C_d = 0.0001$, and one pitching moment count (p.c.) to be $\Delta C_m = 0.00127$.

4.1.5 Investigations

We solve the problem described in this section using three strategies. Each strategy uses the MOO algorithm in Section 3.2, but with different setup of the initial surrogate, as well as with and without the design space reduction step. We compare the strategies in terms of Pareto front findings and computational cost.

www.manaraa.com

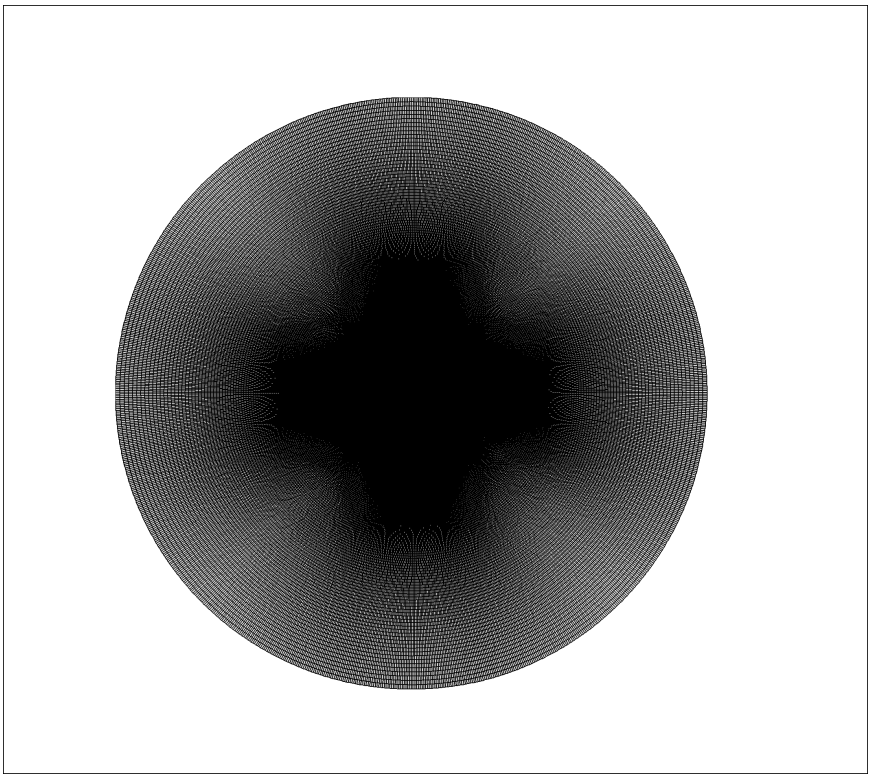

Figure 4.4 Visualization of the high-fidelity model computational mesh.

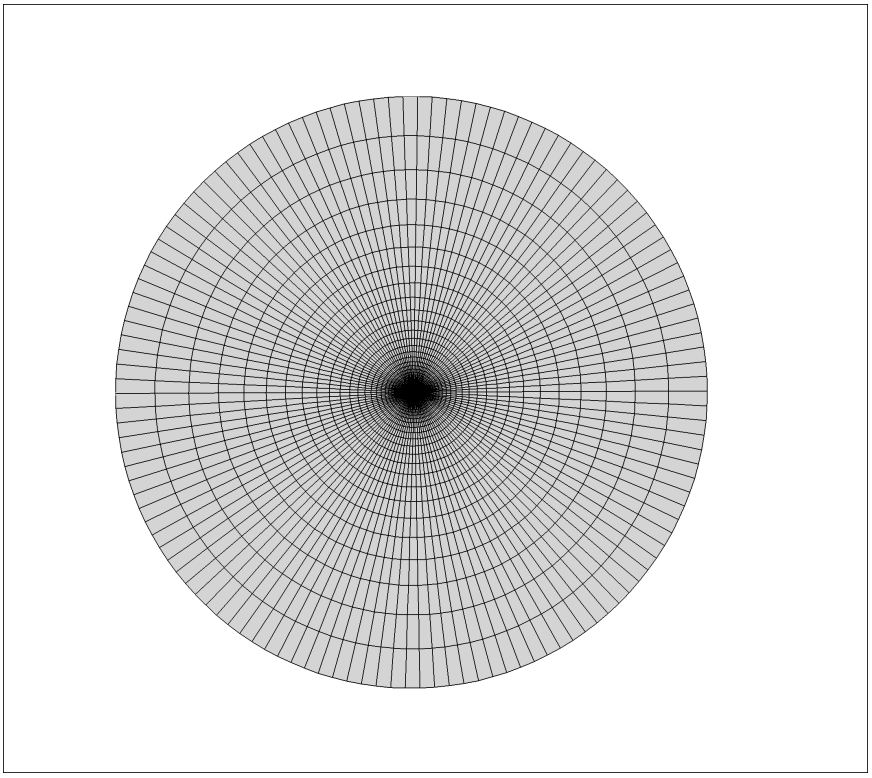

Figure 4.5 Visualization of the low-fidelity model computational mesh.

www.manaraa.com

Table 4.1 Results of the grid convergence study at $M_\infty = 0.734$ and $C_l = 0.824$.

Grid Size	C_d	C_m
64×64	0.0221	0.1384
128×128	0.0228	0.1439
256×256	0.0230	0.1448
512×512	0.0231	0.1450

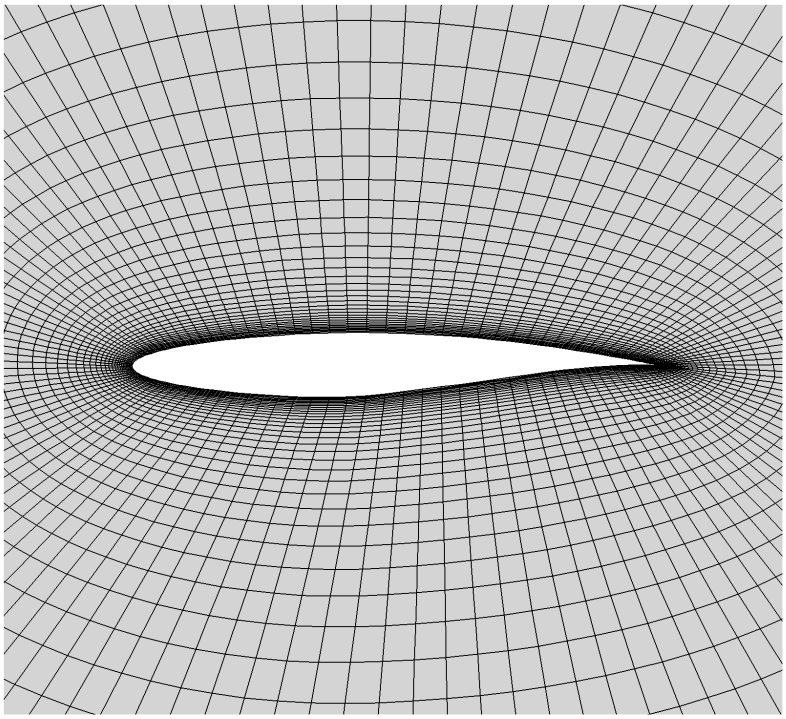

Figure 4.6 A close-up view of the airfoil surface mesh for the high-fidelity model.

www.manaraa.com

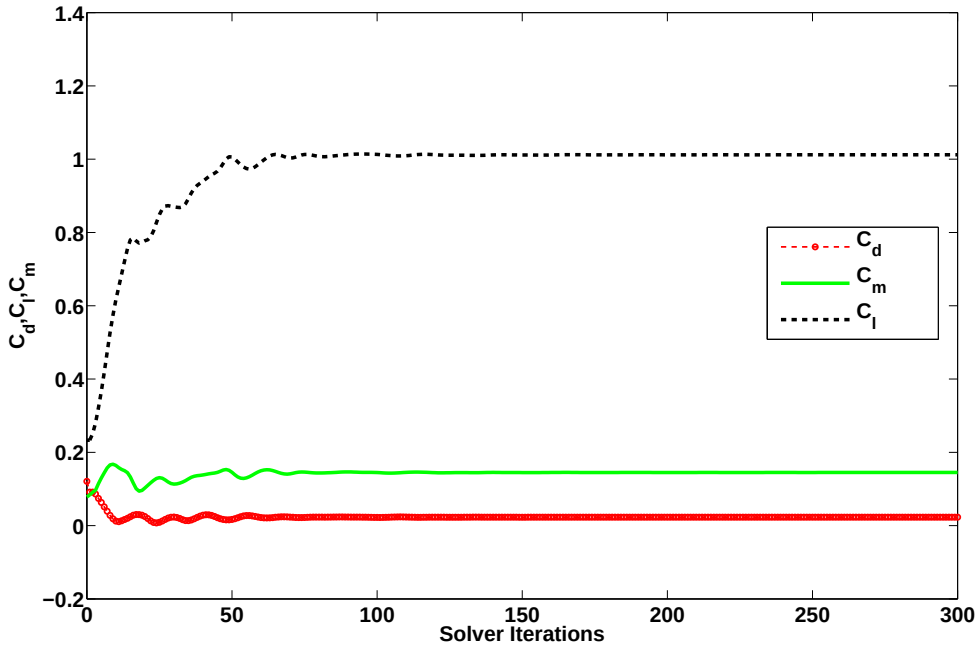

Figure 4.7 Evolution of lift, drag and pitching moment coefficients obtained by the low fidelity model at $M_\infty = 0.734$.

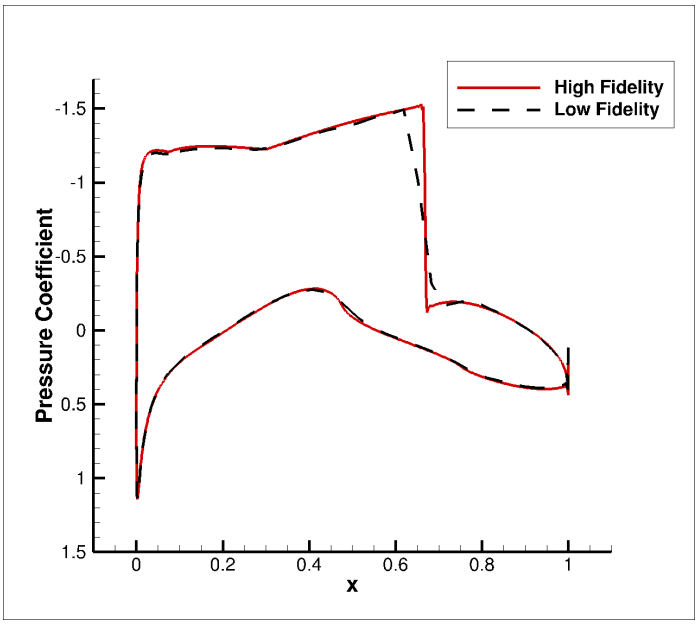

Figure 4.8 A comparison of the pressure distribution obtained by the high and low-fidelity models at $M_\infty = 0.734$.

www.manaraa.com

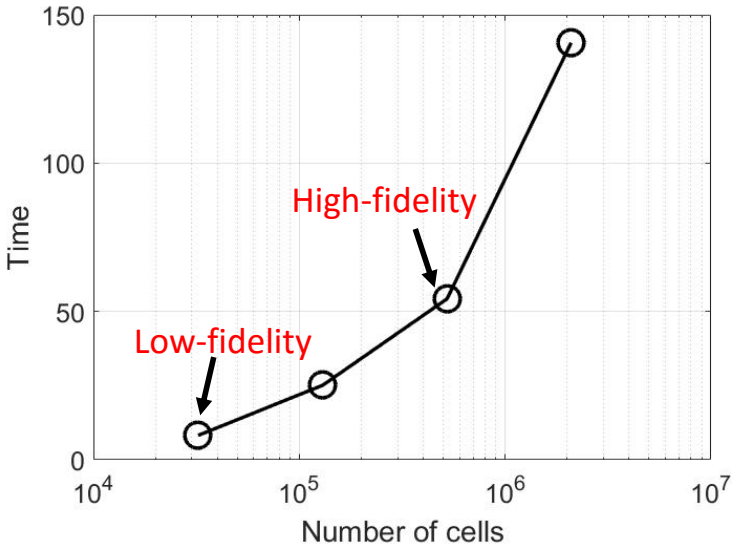

Figure 4.9 Variation of the simulation time with respect to the grid size for the grid study in Table 4.1.

www.manaraa.com

4.2 Strategy 1

In Strategy 1, we follow the MOO algorithm in Section 3.2, but we skip the Output Space Mapping correction and utilize the low-fidelity model, i.e., $\mathbf{s}_0(\mathbf{x}) = \mathbf{c}(\mathbf{x})$. We also skip design space reduction (step 2) and perform MOO on the original design space. Below, we outline the MOO algorithm, give results and summarize the computational cost.

4.2.1 MOO Algorithm

The steps of the MOO algorithm for Strategy 1 are as follows:

1. Sample the design space and acquire the low-fidelity model data with $\mathbf{c}$;
2. Construct a kriging surrogate $\mathbf{s}_{KR}$ based on the data from Step 1;
3. Obtain an approximate Pareto set representation by optimizing $\mathbf{s}_{KR}$ using MOEA;
4. Evaluate the high-fidelity model $\mathbf{f}$ along the Pareto front;
5. Construct/update the co-kriging surrogate $\mathbf{s}_{CO}$;
6. Update Pareto set by optimizing $\mathbf{s}_{CO}$ using MOEA;
7. If termination condition is not satisfied go to Step 5; else END

4.2.2 Results

The MOO algorithm was stopped after 7 iterations. Figure 4.10 shows the 1st, 3rd, 5th, and the 7th Pareto fronts. After the initial Pareto generation using the kriging model, 9 high-fidelity refinement samples are evaluated along the front. Subsequently, the co-kriging model is constructed and optimized using MOEA. This is repeated 7 times. Figure 4.11 shows the final Pareto and several high-fidelity model validation samples. It can be observed that the Pareto front predicted in the final iteration is close to the high fidelity verification samples. However, even after 7 iterations, the agreement between the predicted front and the verification samples is good except at its right-hand-side edge where the actual high-fidelity model samples exhibit higher pitching moments. Further iterations were performed, but it was found that the results did not improve much.

www.manaraa.com

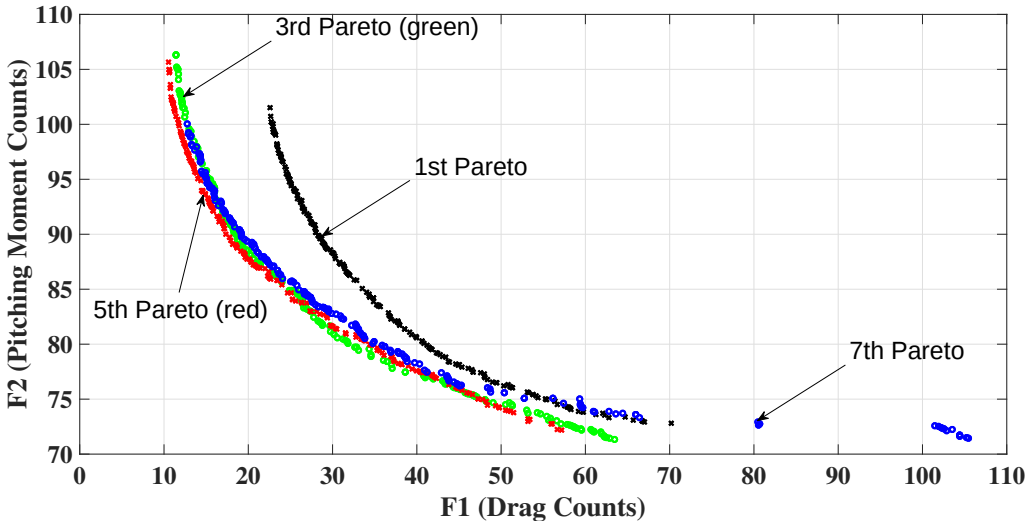

Figure 4.10 Results of Strategy 1 showing the Pareto fronts obtained in at several iterations.

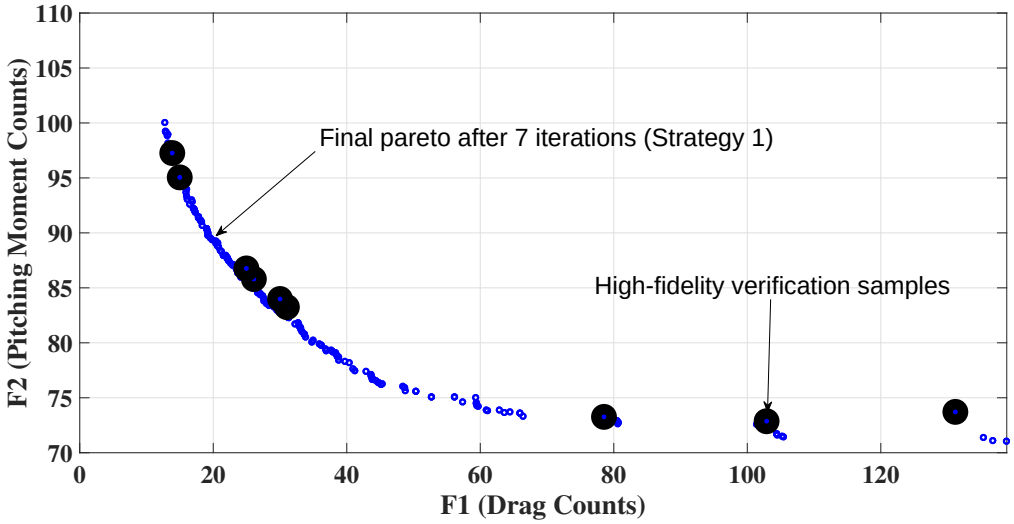

Figure 4.11 The final Pareto front of Strategy 1 with high-fidelity valiation samples.

www.manaraa.com

4.2.3 Computational Cost

The overall computational cost of the multi-objective process in terms of number of evaluations is 1,600 (1,475 feasible points and 125 feasible corner points) low-fidelity model evaluations, and 63 high-fidelity model evaluations (9 each in 7 iterations of co-kriging-based Pareto front refinements). The cost of the kriging and co-kriging function evaluations in the MOEA process is ignored as the calculations are performed very quickly.

In terms of time, there is a significant advantage of the proposed method with variable-fidelity models compared to that of using the high-fidelity directly. Each low-fidelity model simulation is less than 1 min, and each high-fidelity model simulation is around 30 min. Hence, for 1,600 low-fidelity model simulations and 63 high-fidelity model simulations, the total time is around 3,490 minutes, or 2.4 days. However, the proposed method did not converge even after 9 iterations and hence the total time can be up to around 3 days. Using only the high-fidelity model, and assuming 1,663 samples are still needed, the total time would be around 34.6 days.

4.3 Strategy 2

Strategy 2 follows the MOO algorithm in Section 3.2 with the following modifications. We perform the OSM correction of the low-fidelity model (Step 1), but skip the design space reduction (Step 2) and perform MOO on the original design space.

4.3.1 Description

The steps of the MOO algorithm for Strategy 2 are as follows:

1. Correct the low-fidelity model $\mathbf{c}$ using output space mapping to construct a surrogate model $\mathbf{s}_0$;
2. Sample the design space and acquire the surrogate model data with $\mathbf{s}_0$;
3. Construct a kriging surrogate $\mathbf{s}_{KR}$ based on the data from Step 3;
4. Obtain an approximate Pareto set representation by optimizing $\mathbf{s}_{KR}$ using MOEA;
5. Evaluate the high-fidelity model $\mathbf{f}$ along the Pareto front;

www.manaraa.com

6. Construct/update the co-kriging surrogate $\mathbf{s}_{CO}$;

7. Update Pareto set by optimizing $\mathbf{s}_{CO}$ using MOEA;

8. If termination condition is not satisfied go to Step 5; else END

4.3.2 Results

Figure 4.12 shows the results of the Strategy 2 iterations. In this case, three iterations of the MOO are enough to converge. Apart from the 1,600 points, extra 30 feasible points are generated to calculate to construct the initial surrogate $\mathbf{s}_0$. These points include $2n + 1$ corner points (where $n = 8$ is the number of design variables) and the rest points are from LHS sampling (Section 4.1.3). While collecting points to calculate the OSM correction parameters, care is taken that the infeasible designs (the ones violating the area constraint) are removed. The 1,600 low-fidelity data samples are corrected to approximate it near to the high-fidelity model. Using these corrected set, the kriging model $\mathbf{s}_{KR}$ is generated, which is subsequently used to perform the MOEA and get the first Pareto front. Then, the Pareto is refined using 9 high-fidelity verification samples evaluated along the Pareto front to generate the co-kriging model $\mathbf{s}_{CO}$ as described in Section 3.6 until convergence. It can be observed from Fig. 4.12 that the Pareto front predicted in the third iteration is converged within 1 d.c., i.e, after three iterations, the agreement between the predicted front and the refinement samples is withing 1 d.c.

4.3.3 Computational Cost

The overall computational cost of the MOO process in terms of number of evaluations are the initial 1,600 low-fidelity model evaluations, and 30 low- and high-fidelity model evaluations for OSM correction. Then 27 high fidelity model evaluations are used for the refinement phase (9 in each of the 3 iterations). The total time is 3,340 minutes, or 2.3 days.

www.manaraa.com

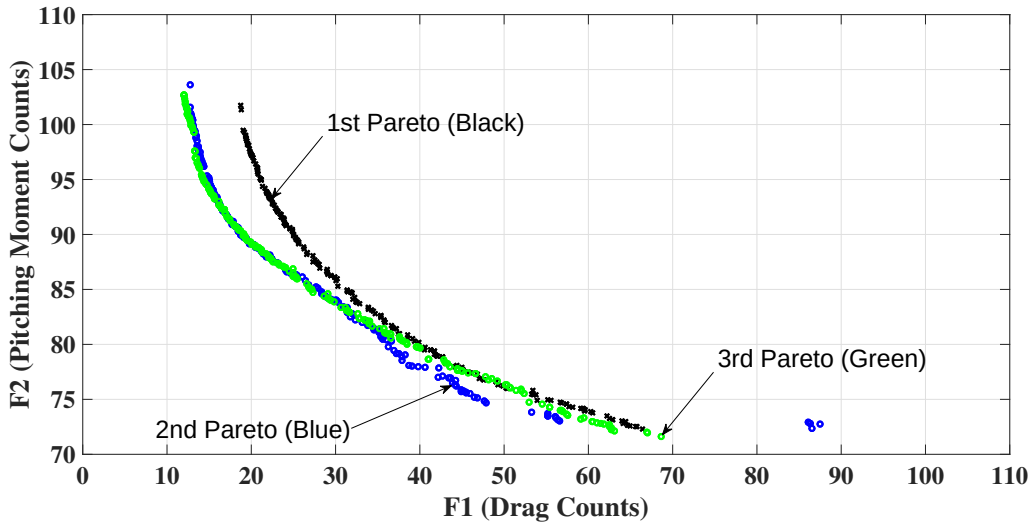

Figure 4.12 Results of Strategy 2 showing the Pareto fronts obtained in at several iterations.

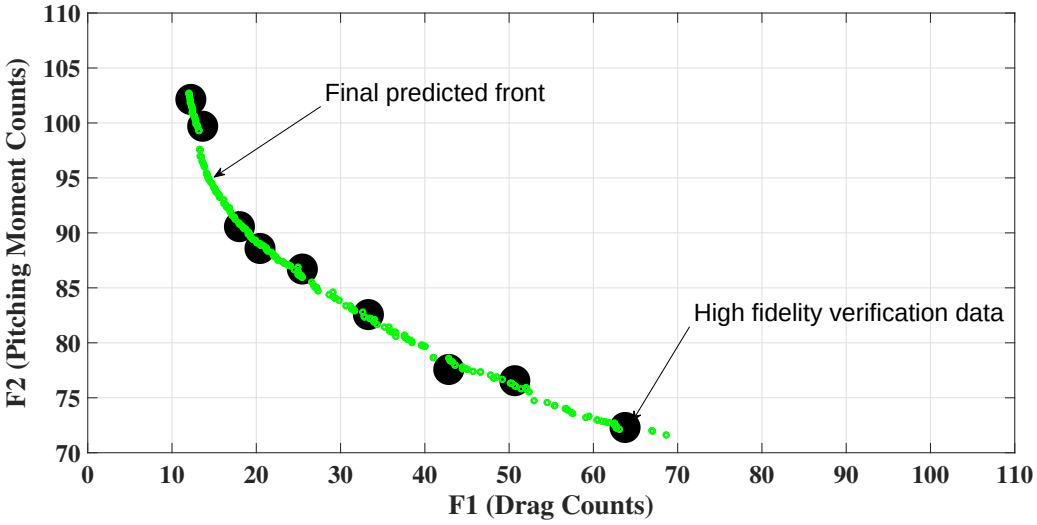

Figure 4.13 The final Pareto front of Strategy 2 with high-fidelity valiation samples.

www.manaraa.com

4.4 Strategy 3

In Strategy 3, we skip the OSM correction in Step 1 of the MOO algorithm in Section 3.2, but perform design space reduction of Step 2.

4.4.1 Description

The steps of the MOO algorithm for Strategy 3 are as follows:

1. Perform design space reduction using $\mathbf{s}_0$;
2. Sample the design space and acquire the low-fidelity model data with $\mathbf{c}$;
3. Construct a kriging surrogate $\mathbf{s}_{KR}$ based on the data from Step 2;
4. Obtain an approximate Pareto set representation by optimizing $\mathbf{s}_{KR}$ using MOEA;
5. Evaluate the high-fidelity model $\mathbf{f}$ along the Pareto front;
6. Construct/update the co-kriging surrogate $\mathbf{s}_{CO}$;
7. Update Pareto set by optimizing $\mathbf{s}_{CO}$ using MOEA;
8. If termination condition is not satisfied go to Step 5; else END

4.4.2 Results

To reduce the design space we perform two single-objective optimization runs to obtain an approximation of the extreme points of the Pareto front. Pattern search method [REF] is used to perform these runs using the low-fidelity CFD model. The optimization runs are performed at $M_\infty = 0.734$ with $A_{baseline} = 0.0779$. To maintain a fixed lift for each new design produced by the optimizer, the angle of attack is used as a dummy parameter. The specification of the single-objective optimization runs are as follows:

$$\mathbf{x}_c^{*(1)} = arg \min_{\mathbf{l} \le \mathbf{x} \le \mathbf{u}} C_{d.c}(\mathbf{x}) \quad \text{s.t.} \quad C_l = 0.824, \quad A \ge A_{baseline},$$

$$\mathbf{x}_c^{*(2)} = arg \min_{\mathbf{l} \le \mathbf{x} \le \mathbf{u}} C_{m.c}(\mathbf{x}) \quad \text{s.t.} \quad C_l = 0.824, \quad A \ge A_{baseline}.$$

www.manaraa.com

Pattern search method needs 256 function evaluations in both optimization runs to reach to the minimum, and optimum designs are:

$$\mathbf{x}_c^{*(1)} = [0.01790, 0.04244, 0.06388, 0.046725, -0.02938, -0.07122, -0.04399, 0.0063],$$
$$C_d^{*(1)} = 0.00239, C_m^{*(1)} = 0.12883,$$

$$\mathbf{x}_c^{*(2)} = [0.02010, 0.042341, 0.06348, 0.03455, -0.02938, -0.07802, -0.04415, 0.00466],$$
$$C_d^{*(2)} = 0.00857, C_m^{*(2)} = 0.08441.$$

The reduced design space (according to Section 3.4) is, therefore, defined by $\mathbf{l}^*$= min($\mathbf{x}_c^{*(1)}$, $\mathbf{x}_c^{*(2)}$), and $\mathbf{u}^*$=max($\mathbf{x}_c^{*(1)}$, $\mathbf{x}_c^{*(2)}$). If we compare $\mathbf{x}_c^{*(1)}$ and $\mathbf{x}_c^{*(2)}$ we find that the 5th design is same for both. It means the value of that design variable is constant at -0.02938 and hence we can reduce the dimension of the design space by removing that design variable. Now the reduced design space has a dimension of 7 instead of 8. Thus, we get the following bounds of the reduced design space:

$$\mathbf{l}^* = [0.0179, 0.0423, 0.0635, 0.0346, -0.0780, -0.0442, 0.0047],$$
$$\mathbf{u}^* = [0.0201, 0.0424, 0.0639, 0.0467, -0.0712, -0.0440, 0.0063].$$

The reduced space is over 100 times smaller (volume-wise) than the original space. In the next step, a kriging interpolation model $\mathbf{s}_{KR}$ is constructed using 1,600 training points, including corner points, and 1,344 samples allocated using LHS. Subsequently, Steps 5-8 of the MOO procedure are executed using 9 high-fidelity verification points uniformly sampled along the Pareto front predicted by the surrogate model. The process was converged in 2 iterations. Figure 4.14 shows the Pareto fronts at subsequent design refinements. In Fig. 4.15 it can be observed that there is a good agreement between the predicted front and the validation samples after 2 iterations, and the algorithm has reached the convergence.

4.4.3 Computational Cost

The total design optimization cost corresponds to 1,856 low-fidelity function evaluations, and 18 high-fidelity refinement samples. The total time is 2,652 min, or 1.8 days.

www.manaraa.com

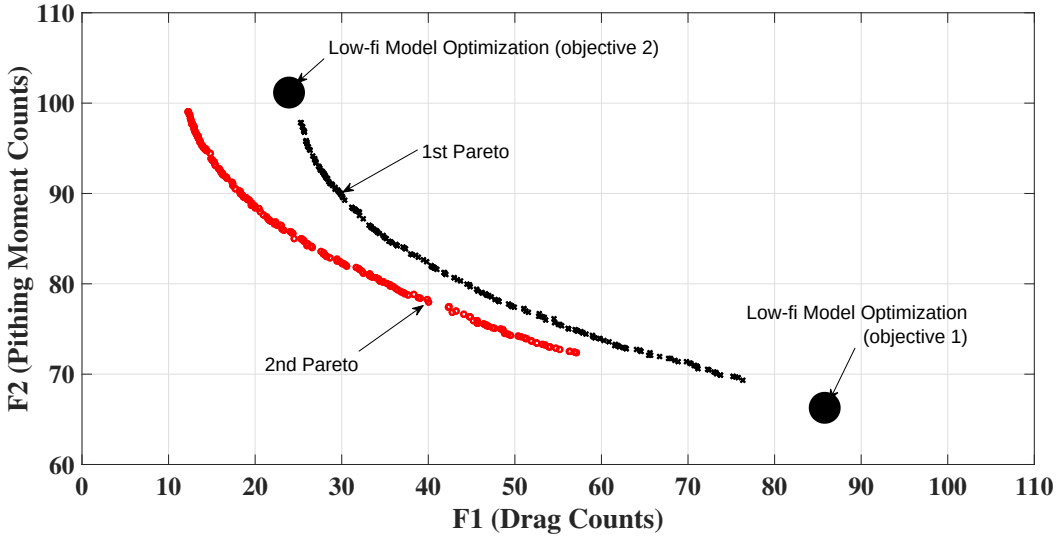

Figure 4.14 Results of Strategy 3 showing the Pareto fronts obtained in at several iterations.

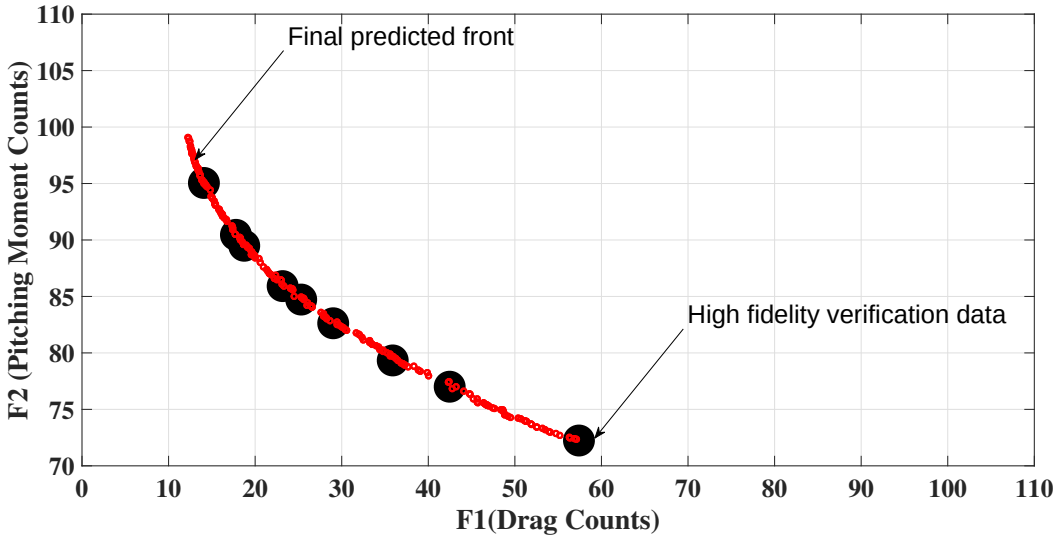

Figure 4.15 The final Pareto front of Strategy 3 with high-fidelity valiation samples.

www.manaraa.com

4.5 Comparison of the Strategies

In this section, we summarize the outcome of each strategy and compare the cost in terms of time and number of evaluations. Figure 4.16 compares the final Pareto fronts for all the three strategies. It can be observed that a majority of the fronts are comparable. Table 4.2 summarizes the cost. Strategy 1 was the least efficient as it needed 3 days. Moreover, Strategy 1 was not fully converged after 7 iterations. Strategy 2 gave good results and converged well within 3 iterations and needed 2.3 days. Strategy 3 was the most efficient method as it converged within 2 iterations and needed 1.8 days. Three designs were selected along the final Pareto front of Strategy 3 (as shown in Fig. 4.17) and the characteristic features are plotted in Figs. 4.18, 4.19, 4.20, 4.21 and 4.22.

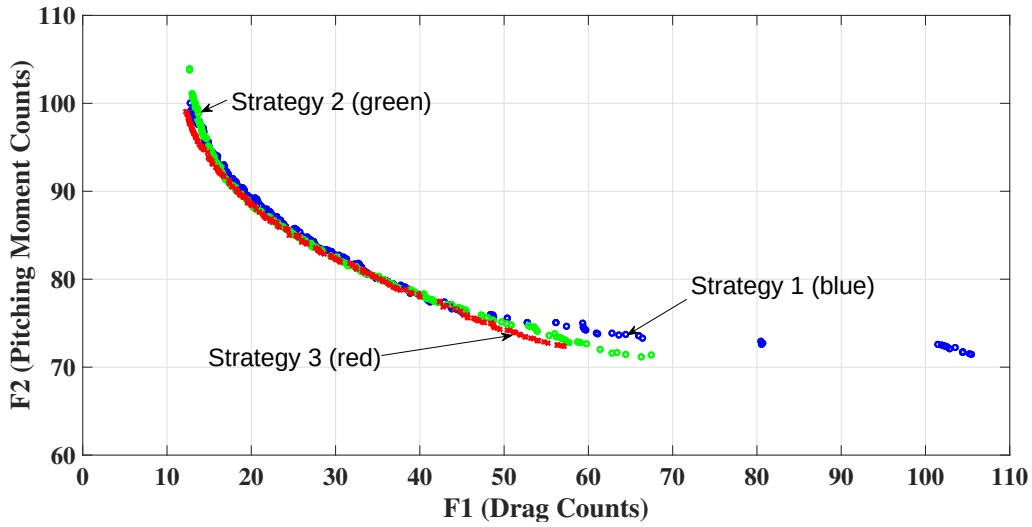

Figure 4.16 Comparision of the final Pareto fronts obtained by the three strategies.

Table 4.2 Comparison of the computatioal cost of the three strategies.

Strategy	Iterations	N_c	N_f	Time (days)
1	7+	1,600	63	3
2	3	1,630	48	2.3
3	2	2,112	9	1.8

www.manaraa.com

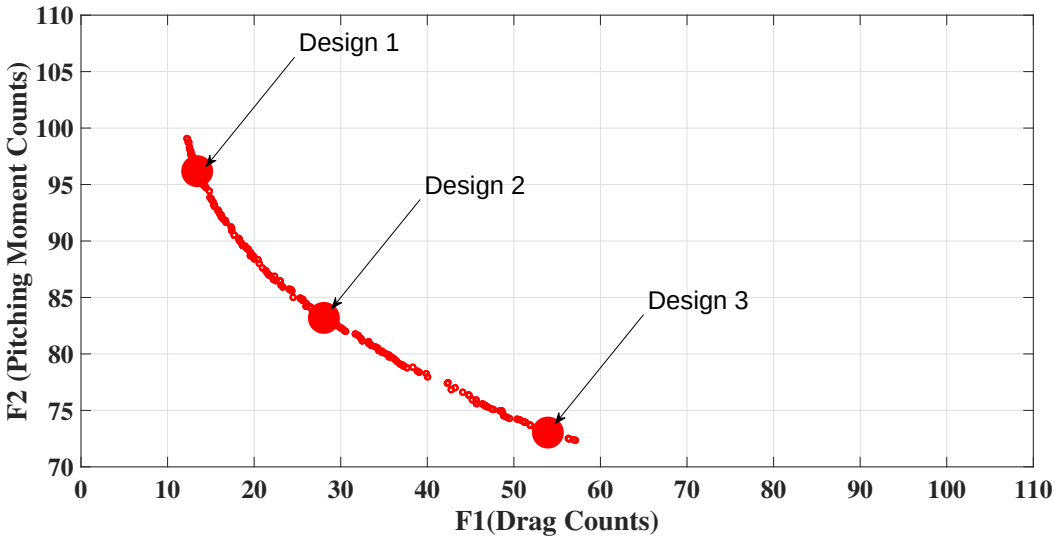

Figure 4.17 Designs selected along the final Pareto front of strategy 3 for visualization.

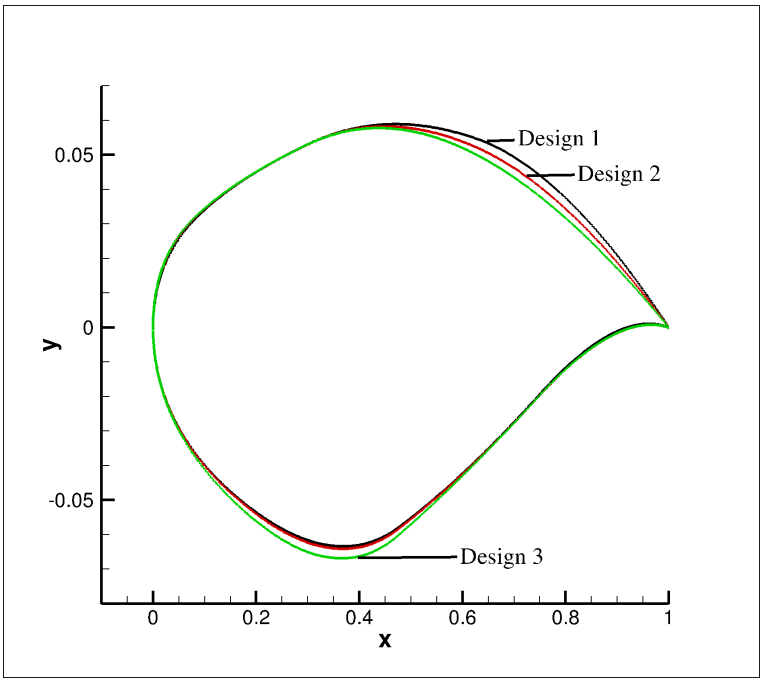

Figure 4.18 Airfoil shape of the designs selected from the final Pareto front of Strategy 3.

www.manaraa.com

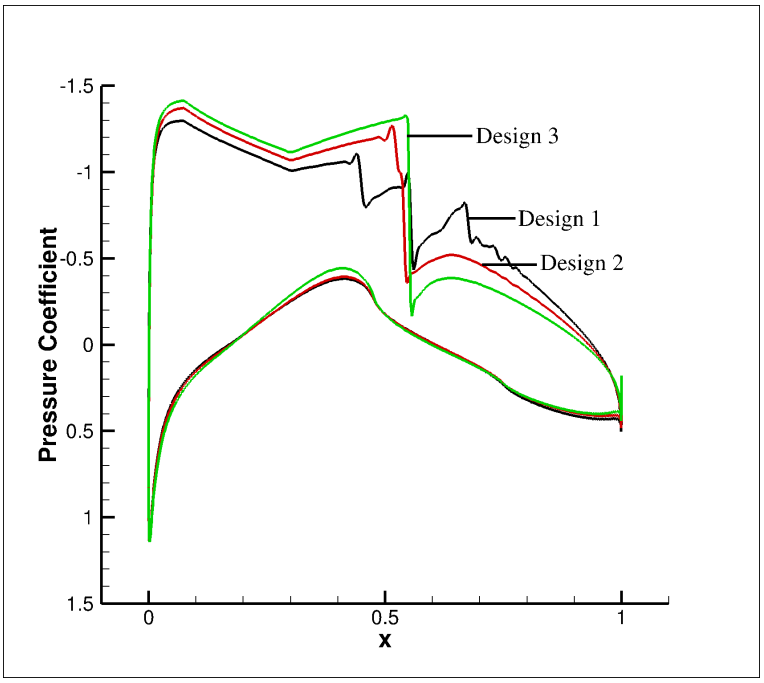

Figure 4.19 Pressure coefficient of designs selected from the final Pareto front of Strategy 3.

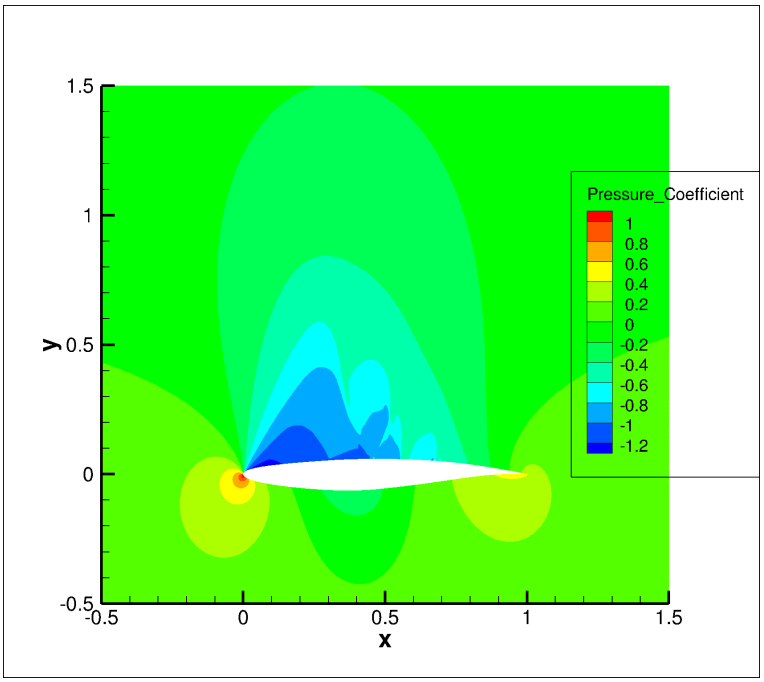

Figure 4.20 Pressure coefficient contours for design 1.

www.manaraa.com

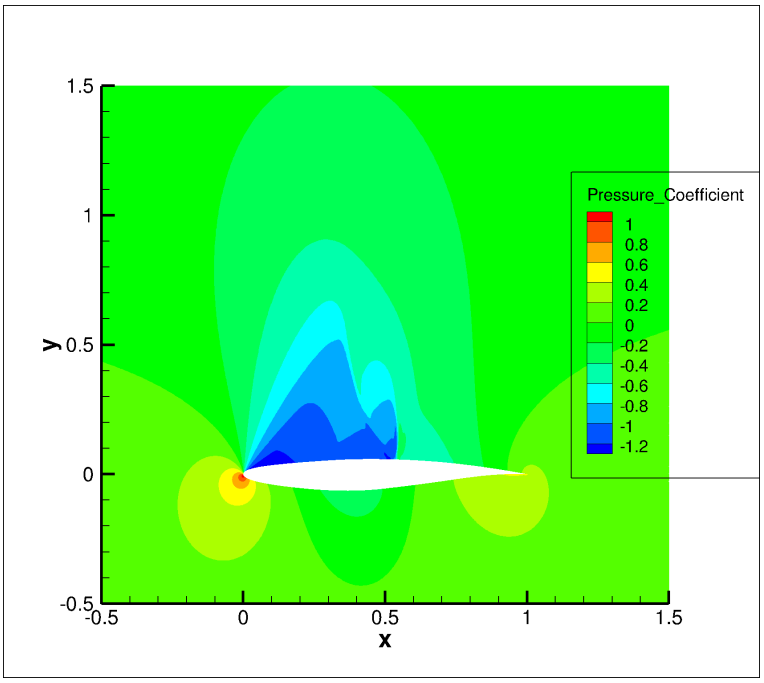

Figure 4.21 Pressure coefficient contours for design 2.

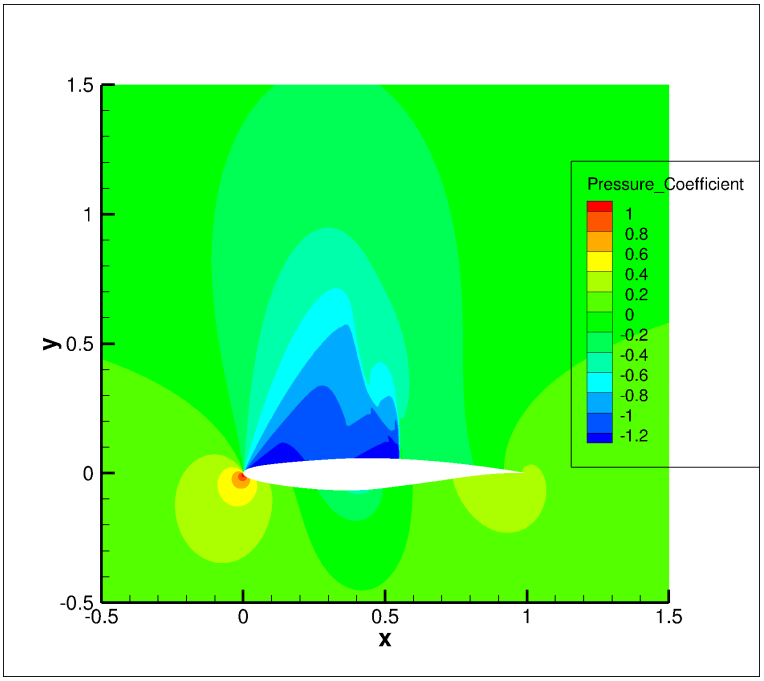

Figure 4.22 Pressure coefficient contours for design 3.

www.manaraa.com

4.6 Comparison using Single-Objective Optimization and a Scalarized Objective

To validate the Pareto fronts obtained by the proposed MOO algorithm, we need to have the Pareto front evaluated entirely using the high-fidelity model. To do this, we compare the results obtained with the single-objective optimization and a scalarized objective function (see Section 2.2.1). In particular, using this approach we find the two extreme points of the Pareto front, and two other points on the front.

The extreme points are found as follows:

$$\mathbf{x}_f^{*(1)} = arg\min_{\mathbf{l}\leq\mathbf{x}\leq\mathbf{u}} C_{d.f}(\mathbf{x}) \quad \text{s.t.} \quad C_l = 0.824, \quad A \geq A_{baseline},$$

$$\mathbf{x}_f^{*(2)} = arg\min_{\mathbf{l}\leq\mathbf{x}\leq\mathbf{u}} C_{m.f}(\mathbf{x}) \quad \text{s.t.} \quad C_l = 0.824, \quad A \geq A_{baseline}.$$

And the results are:

$$\mathbf{x}_f^{*(1)} = [0.01790, 0.04244, 0.06388, 0.04672, -0.02938, -0.07122, -0.04399, 0.0063],$$
$$C_{d.f}^{*(1)} = 0.0011, \quad C_{m.f}^{*(1)} = 0.1319,$$

$$\mathbf{x}_f^{*(2)} = [0.02010, 0.042341, 0.06348, 0.03455, -0.02928, -0.07802, -0.04415, 0.0046],$$
$$C_{d.f}^{*(2)} = 0.0073, \quad C_{d.f}^{*(2)} = 0.0862.$$

The other two points are found as:

$$\mathbf{x}_f^{*(3\ \&\ 4))} = arg\min_{\mathbf{l}\leq\mathbf{x}\leq\mathbf{u}} w_1 \times C_{d.f}(\mathbf{x}) + w_2 \times C_{m.f}(\mathbf{x}) \quad \text{s.t.} \quad C_l = 0.824, \quad A \geq A_{baseline},$$

where the weights w_1 and w_2 are varied to get different points along the front. For each case, we use $w_1 = [0.2 \quad 0.75]$ and $w_2 = [0.8 \quad 0.25]$. The results are as follows:

$$\mathbf{x}_f^{*(3)} = [0.02010, 0.04230, 0.06534, 0.03615, -0.02934, -0.07522, -0.04415, 0.0046],$$
$$C_{d.f}^{*(3)} = 0.0050, C_{m.f}^{*(3)} = 0.0941,$$
$$\mathbf{x}_f^{*(4)} = [0.01823, 0.04230, 0.06388, 0.04619, -0.02491, -0.07076, -0.04409, 0.00463],$$
$$C_{d.f}^{*(4)} = 0.0012, C_{m.f}^{*(4)} = 0.1230.$$

www.manaraa.com

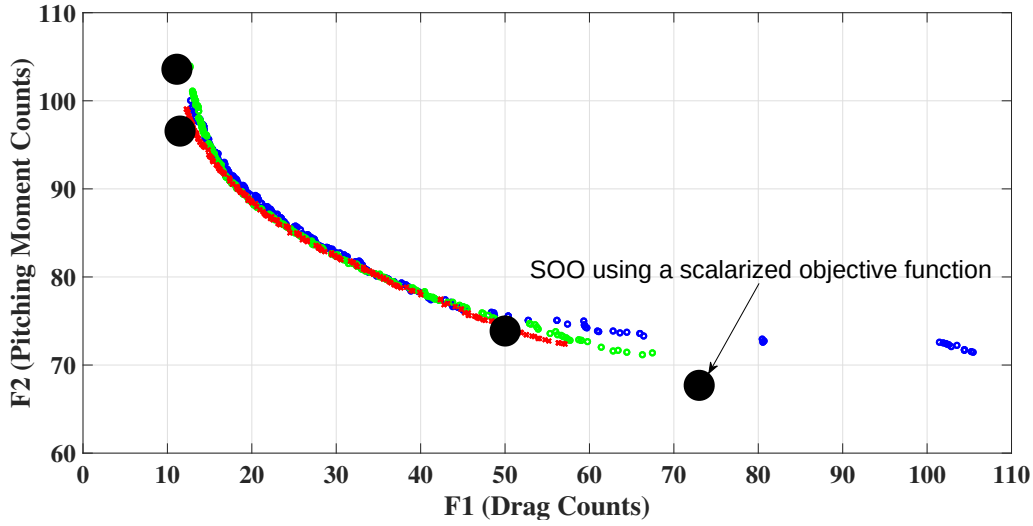

Figure 4.23 A comparison of the proposed multi-objective algorithm with the single-objective optimization using a scalarized objective function.

Figure 4.23 show a comparison of the Pareto fronts obtained by the proposed MOO algorithm and the single-objective optimization (SOO). It can be seen that the final Pareto fronts compare well with the SOO results. Although, it seems that the proposed MOO algorithm does not give the full front.

4.7 Parametric Study of Strategy 3

In this section, we perform a parametric study of Strategy 3 where the number of low-fidelity model sampling points. In each case, we re-run the MOO algorithm and compare with the initial results.

4.7.1 Description

Previously, we used 1,600 low-fidelity points. Here, we use 500, 300, 100, and 50 sampling points. The relative root mean square error (RMSE) is calculated, as described in Section 3.5.3, with 5 sets of samples: 1,600, 500, 300, 100, 50; all sampled uniformly over the reduced design space. The kriging model was constructed using each of them, and the relative RMSE was calculated.

www.manaraa.com

4.7.2 Results

The relative RMSE for the initial kriging model, generated during Step 4 of the MOO algorithm in Strategy 3 and Strategy 1, for different number of samples is given in Table 4.3 and Table 4.4 respectively. It can be seen that the RMSE is very low for the 1,600 samples (or 0.17%) in Strategy 3, indicating that the sampling number is too high. At 50 samples, the RMSE is 3.4%, which is higher than the recommended value of 2% (as mentioned in Section 3.5.3). The 100 samples, with a relative RSME value of 1.1%, seem to be adequate in this particular case. The relative RMSE value for the 1,600 samples in the full design space (used in Strategy 1) is around 1.45%. This fact, and that Strategies 1 and 3 yield similar results, indicate that a relative RSME value of around 1% is adequate in this particular case. The results of the MOO runs for cases with kriging models generated by 500, 300, and 100 low-fidelity model samples are given in Figs. 4.24, 4.25, and 4.26, respectively. All the runs converge after 2 iterations, and the final Pareto fronts compare well with the validation high-fidelity samples.

Further investigation is performed to check for a minimum number of high-fidelity refinement samples. For this test, 100 low-fidelity samples are used to construct the kriging model. Instead of 9 refinement points for the co-kriging surrage, we try 3 refinement points. The points are chosen in a way that we have 2 extreme points of the initial Pareto front, and one point in the middle of the front. Figure 4.27 shows the results. Again, the MOO algorithm converges within 2 iterations and compares well with the validation samples. Figure 4.28 compares this Pareto front with the one obtained when using 1,600 samples, and indicates a very good agreement of the fronts. The overall computational time with 100 initial samples and 3 high-fidelity refinement points is approximately 0.5 days. Table 4.5 give a comparison with the original results.

www.manaraa.com

Table 4.3 Relative root mean square error (RMSE) of the initial kriging model in Step 4 of the MOO algorithm in Strategy 3 for different number of samples.

Samples	Relative RMSE (%)
1,600	0.17
500	0.25
300	0.50
100	1.13
50	3.43

Table 4.4 Relative root mean square error (RMSE) of the initial kriging model in Step 4 of the MOO algorithm in Strategy 1 for different number of samples.

Samples	Relative RMSE (%)
1,600	1.45
500	2.62
300	4.50
100	6.04

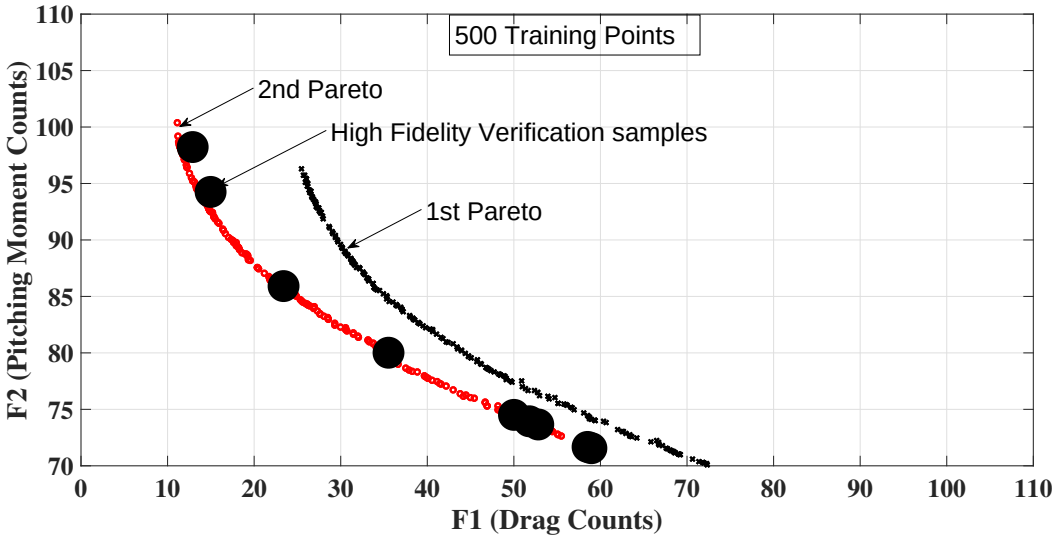

Figure 4.24 Results of Strategy 3 showing the Pareto fronts obtained with 500 initial sampling points.

www.manaraa.com

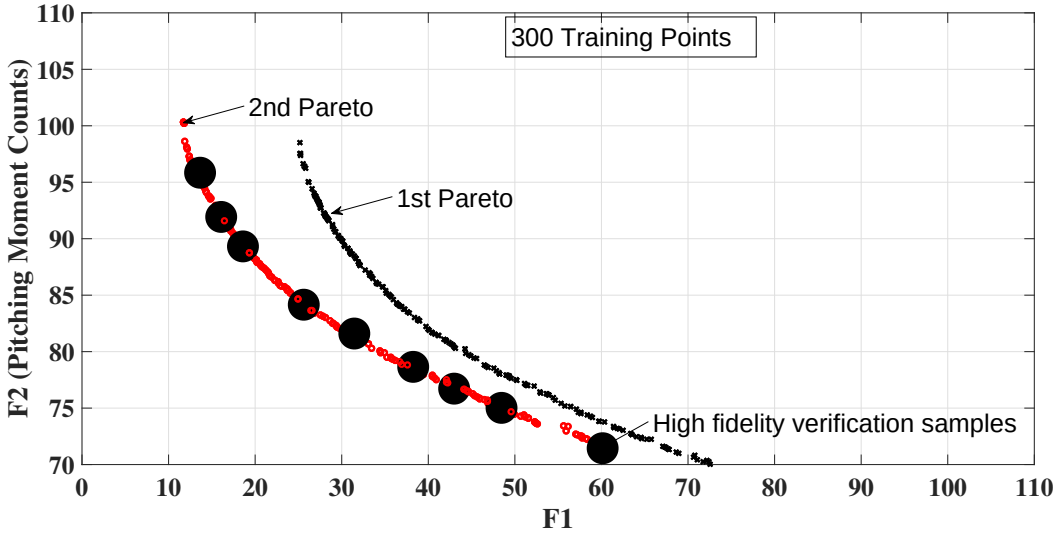

Figure 4.25 Results of Strategy 3 showing the Pareto fronts obtained with 300 initial sampling points.

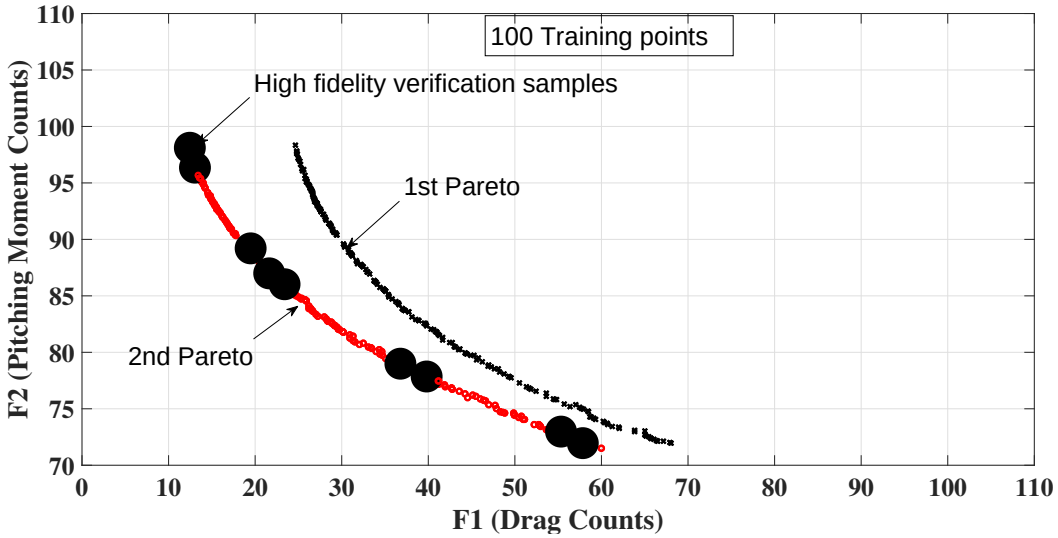

Figure 4.26 Results of Strategy 3 showing the Pareto fronts obtained with 100 initial sampling points.

www.manaraa.com

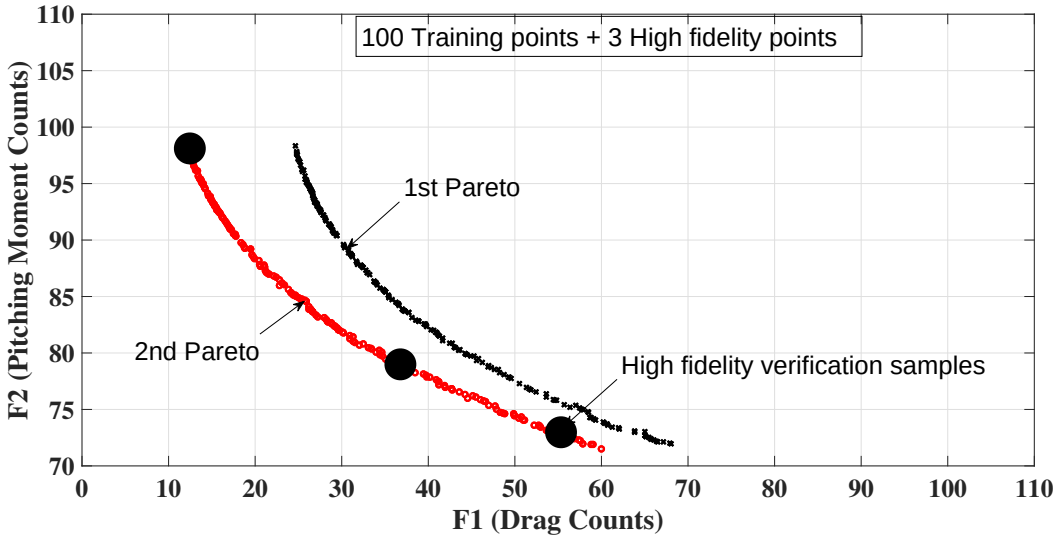

Figure 4.27 Results of Strategy 3 showing the Pareto fronts obtained with 100 initial sampling points and 3 refinement points.

www.manaraa.com

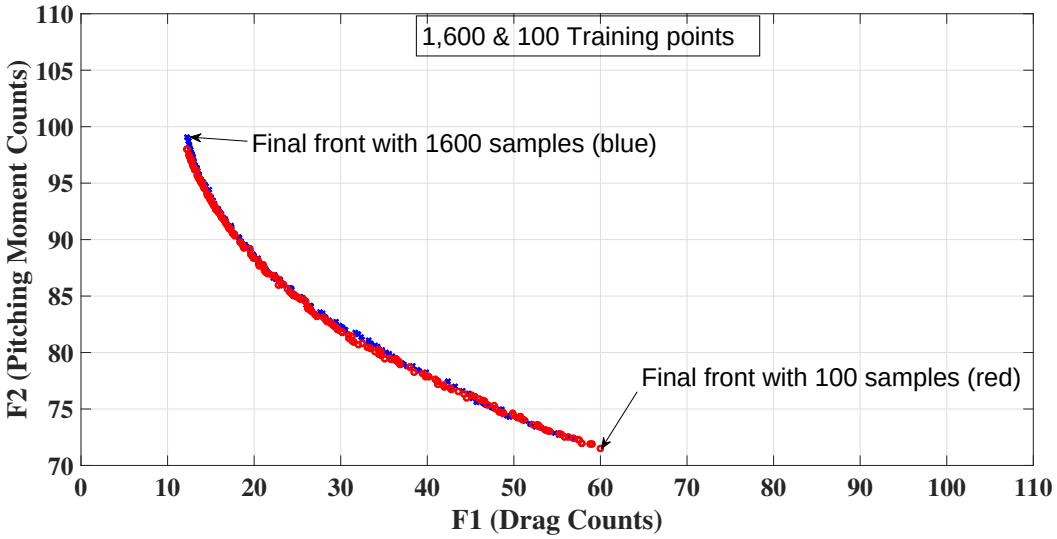

Figure 4.28 Results of Strategy 3 showing the Pareto fronts obtained with 1,600 and 100 initial sampling points.

Table 4.5 Comparison of the computatioal cost of the original three strategies, with the refined Strategy 3 (indicated by 3*).

Strategy	**Iterations**	$\mathbf{N_c}$	$\mathbf{N_f}$	**Time (days)**
1	7+	1,600	63	3
2	3	1,630	48	2.3
3	2	2,112	9	1.8
3*	2	612	3	0.5

www.manaraa.com

CHAPTER 5. CONCLUSION

A procedure for expedited multi-objective optimization (MOO) of aerodynamic surface has been presented. The approach exploits design space reduction, variable-fidelity computational fluid dynamics (CFD) simulations, and kriging and co-kriging interpolation models to construct accurate and fast surrogate models. The latter is utilized to generate (by evolutionary optimization) a set of alternative solutions representing the best possible trade-offs between conflicting objectives. The design space reduction applied at the beginning of the optimization process allows for identifying the region of the search space that contains Pareto optimal solutions. The application of the MOO algorithm is demonstrated on the design of airfoil shapes in two-dimensional transonic inviscid flow. The results the MOO algorithm in the reduced space compare well with results obtained when considering the full design space, as well as when using the weighted sum method. The proposed MOO approach was demonstrated to be at least an order of magnitude faster than using the high-fidelity model directly.

The demonstration case involved a low-dimensional parameterization (8 design variables) and a small scale simulation (a high-fidelity simulation with over 512,000 unknowns, requiring around 30 min on HPC). Future work will be focused on extending the approach for medium- to high-dimensional cases (where the number of design variables range from 10 to 40, or higher, as well as larger scales of simulations. In these cases, the initial computational effort related to the construction of the kriging and co-kriging models may become a serious issue (due to the curse of dimensionality) despite of the design space reduction step used in the present version of the algorithm. It is likely that in such cases a reduction in the design space dimensionality needs to be performed prior to the design space reduction process (which focuses mainly on the ranges of the parameters). The dimensionality reduction can be performed using principal component analysis (PCA). In that case, adjoint sensitivity information needs to be acquired.

www.manaraa.com

BIBLIOGRAPHY

[1] S. Koziel, A. Bekasiewicz, and L. Leifsson, "Multi-objective optimization of planar yagi-uda antenna using physics-based surrogates and rotational design space reduction," *Int. Conf. Comp. Science, Reykjavik, Iceland*, June 1–3, 2015.

[2] C. Fonseca, *Multiobjective genetic algorithms with applications to control engineering problems*. PhD thesis, Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, UK, 1995.

[3] C. L. Hwang and A. S. M. Masud, *Multiple objective decision making, methods and applications: a state-of-the-art survey*. Berlin, Germany: Springer-Verlag, 1947.

[4] K. Miettinen, *Nonlinear Multiobjective Optimization*. Boston, MA: Kluwer Academic Publishers, 1999.

[5] J. Holland, *Adaptation in Natural and Articial Systems*. Ann Harbour, Michigan: The University of Michigan Press, 1975.

[6] E. Zitzler, "Evolutionary algorithm for multi-objective optimization:methods and application," Master's thesis, Institut fur Technische Informatik und Kommunikationsnetze Computer Engineering and Networks Laboratory, Zurich, Switzerland, 1999.

[7] R. C. Eberhart and Y. Shi, "Comparison between genetic algorithms and particle swarm optimization," *Evolutionary Programming VII, Springer Berlin, Germany*, pp. 611–616, March 1998.

[8] T. Simpson, J. Peplinski, P. Koch, and J. Allen, "Metamodels for computer-based engineering design: survey and recommendations," *Engineering with Computers*, vol. 17, no. 2, pp. 129–150, 2001.

www.manaraa.com

[9] C. Huang, J. Galuski, and C. Bloebaum, "Multi-objective pareto concurrent subspace optimization for multidisciplinary design," *AIAA journal*, vol. 45, no. 8, 2007.

[10] V. Pareto, *Manuale di Economica Politica and Societa Editrice Libraria. Milan.* New York, NY: A.M. Kelley, 1971.

[11] T. Athan and P. Papalambros, "A note on weighted criteria methods for compromise solutions in multi-objective optimization," *Eng. Optim.*, vol. 27, pp. 155–176, 1996.

[12] A. Geoffrion, "Proper eciency and the theory of vector maximization," *J. Math. Anal. Appl.*, vol. 22, pp. 618–630, 1968.

[13] P. Yu, *Multiple-Criteria Decision Making Concepts, Techniques, and Extensions.* New York, NY: Plenum Press, 1985.

[14] I. Yong Kim and O. de Weck, "Adaptive weighted sum method for multiobjective optimization," *Structural and Multidisciplinary Optimization*, vol. 29, pp. 149–158, 2005.

[15] T. Back, U. Hammel, and H. Schwefel, "Evolutionary computation: Comments on the history and current state," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 3–17, 1997.

[16] C. M. Fonseca and P. Fleming, "An overview of evolutionary algorithms in multiobjective optimization," *Evolutionary Computation*, vol. 3, no. 1, pp. 1–16, 1995.

[17] J. D. Schaffer, *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms.* PhD thesis, Vanderbilt University, Nashville, TN, 1984.

[18] J. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," *The First International Conference on Genetic Algorithms and their Applications, Pittsburgh, PA*, pp. 93–100, 1985.

[19] C. M. Fonseca and P. J. Fleming, "Multiobjective optimization and multiple constraint handling with evolutionary algorithmspart ii: Application example," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 28, no. 1, pp. 38–47, 1998.

www.manaraa.com

[20] J. Kocer, F.Y.and Arora, "Optimal design of h-frame transmission poles for earthquake loading," *J. Struct. Eng.*, vol. 125, pp. 1299–1308, 1999.

[21] M. Gen and R. Cheng, *Genetic Algorithms and Engineering Design*. New York, NY: John Wiley and Sons, 1997.

[22] L. Davis, *Handbook of Genetic Algorithms*. New York, NY: Van Nostrand Reinhold, 1991.

[23] J. Moore and R. Chapman, "Application of particle swarm to multiobjective optimization," *Department of Computer Science and Software Engineering, Auburn University, Auburn, Alabama*, 1999.

[24] C. A. C. Coello, D. A. V. Veldhuizen, and G. B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*. Boston, MA: Kluwer Academic Publishers, May 2002.

[25] M. Reyes-Sierra and C. A. C. Coello, "Multi-objective particle swarm optimizers: A survey of the state of the art," *International Journal of Computational Intelligence Research*, vol. 2, no. 1, pp. 287–308, 2006.

[26] K. Leoviriyakit, S. Kim, and A. Jameson, "Viscous aerodynamic shape optimization of wings including planform variables," *21st Applied Aerodynamics Conference, Orlando, Florida*, June 23–26, 2003.

[27] R. Braembussche, *Numerical Optimization for Advanced Turbomachinery Design, Optimization and Computational Fluid Dynamics*. Berlin, Germany: Springer verlag, 2008.

[28] C. Mader and J. R. R. A. Martins, "Derivatives for time-spectral computational fluid dynamics using an automatic differentiation adjoint," *AIAA Journal*, vol. 50, no. 12, pp. 2809–2819, 2012.

[29] B. Epstein and S. Peigin, "Constrained aerodynamic optimization of three-dimensional wings driven by navier-stokes computations," *AIAA Journal*, vol. 43, no. 9, pp. 1946–1957, 2005.

[30] J. Nocedal and S. Wright, *Numerical Optimization*. New York, NY: Springer, 2006.

www.manaraa.com

[31] S. Kim, K. Hosseini, K. Leoviriyakit, and A. Jameson, "Enhancement of class of adjoint design methods via optimization of parameters," *AIAA Journal*, vol. 48, no. 6, pp. 1072–1076, 2010.

[32] S. Schmidt, N. Gauger, C. Ilic, and V. Schulz, "Three dimensional large scale aerodynamic shape optimization based on shape calculus," *41st AIAA Fluid Dynamics Conference and Exhibit, AIAA Paper 2011-3718, Honolulu, Hawaii*, June 27-30, 2011.

[33] N. Queipo, R. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P. Tucker, "Surrogate-based analysis and optimization," *Progress in Aerospace Sciences*, vol. 41, no. 1, pp. 1–28, 2005.

[34] A. Booker, J. Dennis Jr., P. Frank, D. Serafini, V. Torczon, and M. Trosset, "A rigorous framework for optimization of expensive functions by surrogates," *Structural Optimization*, vol. 17, no. 1, pp. 1–13, 1999.

[35] A. Forrester and A. Keane, "Recent advances in surrogate-based optimization," *Progress in Aerospace Sciences*, vol. 45, no. 1–3, pp. 50–79, 2009.

[36] S. Koziel, D. Echeverra-Ciaurri, and L. Leifsson, *Surrogate-based methods, in S. Koziel and X.S. Yang (Eds.) Computational Optimization, Methods and Algorithms, Series: Studies in Computational Intelligence*. Berlin, Germany, Springer-Verlag, 2011.

[37] S. Guo, "Aeroelastic optimization of an aerobatic aircraft wing structure," *Aerospace Science and Technology*, vol. 11, no. 5, pp. 396–404, 2007.

[38] W. W.Li and C. Pak, "Aeroelastic optimization study based on the x-56a model," *AIAA Aviation, Atlanta, GA*, June 16–20, 2014.

[39] A. Kai, G. J. Kennedy, and J. Martins, "Concurrent aerostructural topology optimization of a wing box," *Computers and Structures*, vol. 134, pp. 1–17, 2014.

[40] G. K. Kenway, J. Martins, and G. Kennedy, "Aerostructural optimization of the common research model configuration," *Group (ADODG)*, vol. 6, no. 7, pp. 8–9, 2014.

www.manaraa.com

[41] E. M. Alfayyadh, S. H. Bakhy, and Y. M. Shkara, "A new multi-objective evolutionary algorithm for optimizing the aerodynamic design of hawt rotor," *ASME 2014 12th Biennial Conference on Engineering Systems Design and Analysis, Copenhagen, Denmark*, vol. 2, July 25-27, 2014.

[42] H. M. Omara and M. Abidob, "Designing integrated guidance law for aerodynamic missiles by hybrid multi-objective evolutionary algorithm and tabu search," *Aerospace Science and Technology*, vol. 14, no. 5, pp. 356–363, 2010.

[43] R. Mukesha, R. Pandiyarajanb, U. Selvakumarc, and D. Lingadurai, "Influence of search algorithms on aerodynamic design optimisation of aircraft wings," *International Conference On Modelling Optimization And Computing*, vol. 38, pp. 2155–2163, 2012.

[44] B. Beachkofski and R. Grandhi, "Improved distributed hypercube sampling," *AIAA Paper 2002-1274, 43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Denver, CO*, April 22–25, 2002.

[45] C. Coello Coello and G. Lamont, *Applications of Multi-Objective Evolutionary Algorithms*. Singapore, Singapore: World Scientific, 2004.

[46] S. Koziel, Q. Cheng, and J. Bandler, "Space mapping," *IEEE Microwave Magazine*, vol. 9, no. 6, pp. 105–122, 2008.

[47] S. Koziel and L. Leifsson, "Knowledge-based airfoil shape optimization using space mapping," *AIAA Paper 2012-3016, 30th AIAA Applied Aerodynamics Conference, New Orleans, Louisiana*, June 25-28, 2012.

[48] J. Kleijnen, *Design and Analysis of Simulation Experiments*. Switzerland: Springer International Publishing, 2015.

[49] J. Sacks, W. Welch, T. Mitchell, and H. Wynn, "Design and analysis of computer experiments," *Statistical Science*, vol. 4, no. 4, pp. 409–435, 1989.

www.manaraa.com

[50] S. Koziel, I. Ogurtsov, S.and Couckuyt, and T. Dhaene, "Variable-fidelity electromagnetic simulations and co-kriging for accurate modeling of antennas," *IEEE Trans. Antennas Prop.*, vol. 61, no. 3, pp. 1301–1308, 2013.

[51] M. Morris, T. Mitchell, and D. Ylvisaker, "Design and analysis of computer experiments: use of derivatives in surface prediction," *Technometrics*, vol. 35, no. 3, pp. 243–255, 1993.

[52] S. Koziel, A. Bekasiewicz, and L. Leifsson, "Multi-objective optimization of planar yagi-uda antenna using physics-based surrogates and rotational design space reduction," *Int. Conf. Comp. Science, Reykjavik, Iceland*, 2015.

[53] A. Forrester, A. Sobester, and A. Keane, "Multi-fidelity optimization via surrogate modelling," *Royal Society, Proceedings of the royal Society*, vol. 463, no. 2088, pp. 3251–3269, 2007.

[54] G. Farin, *Curves and Surfaces for Computer Aided Geometric Design*. Boston, MA: Academic Press, 1993.

[55] F. Palacios, M. R. Colonno, A. C. Aranake, A. Campos, S. R. Copeland, T. D. Economon, A. K. Lonkar, T. W. Lukaczyk, T. W. R. Taylor, and J. J. Alonso, "Stanford university unstructured (su2): An open-source integrated computational environment for multi-physics simulation and design," *AIAA Paper 2013-0287, 51st AIAA Aerospace Sciences Meeting and Exhibit, Grapevine, Texas, USA*, 2013.

[56] A. Jameson, W. Schmidt, and E. Turkel, "Numerical solution of the euler equations by finite volume methods using runge-kutta time-stepping schemes," *AIAA 1981-1259, AIAA 14th Fluid and Plasma Dynamic Conference, Palo Alto, CA*, June 23-25, 1981.

www.manaraa.com