

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/317443697>

# Design strategies for multi-objective optimization of aerodynamic surfaces

Article in *Engineering Computations* · June 2017

DOI: 10.1108/EC-07-2016-0239

CITATIONS

12

READS

112

3 authors:



**Anand Amrit**

Iowa State University

18 PUBLICATIONS 148 CITATIONS

[SEE PROFILE](#)



**Leifur Leifsson**

Iowa State University

226 PUBLICATIONS 2,341 CITATIONS

[SEE PROFILE](#)



**Slawomir Koziel**

Reykjavik University

988 PUBLICATIONS 11,334 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Optimization of power electrical distribution network structures for improved power supply reliability [View project](#)



Dynamic Stall on an Aircraft [View project](#)

Citation

Amrit, A., Leifsson, L. and Koziel, S. (2017),  
"Design strategies for multi-objective optimization of aerodynamic surfaces", Engineering Computations,  
Vol. 34 No. 5, pp. 1724-1753. <https://doi.org/10.1108/EC-07-2016-0239>

# Design Strategies for Multi-Objective Optimization of Aerodynamic Surfaces

## Structured Abstract

### Purpose

This paper investigates several design strategies to solve multi-objective aerodynamic optimization problems using high-fidelity simulations. The purpose is to find strategies which reduce the overall optimization time while still maintaining accuracy at the high-fidelity level.

### Design/methodology/approach

Design strategies are proposed that utilize an algorithmic framework composed of (1) search space reduction, (2) fast surrogate models constructed using a combination of physics-based surrogates and kriging, and (3) global refinement of the Pareto front with co-kriging. The strategies either search the full or reduced design space with a low-fidelity model or a physics-based surrogate.

### Findings

Numerical investigations of airfoil shapes in two-dimensional transonic flow are used to characterize and compare the strategies. The results show that searching a reduced design space produces the same Pareto front as when searching the full space. Moreover, since the reduced space is two orders of magnitude smaller (volume-wise), the number of required samples to setup the surrogates can be reduced by an order of magnitude. Consequently, the computational time is reduced from over three days to less than half a day.

### Originality/value

The proposed design strategies are novel and holistic. The strategies render multi-objective design of aerodynamic surfaces using high-fidelity simulation data in moderately sized search spaces computationally tractable.

**Keywords:** aerodynamic shape optimization, multi-objective design, variable-fidelity CFD models, design space reduction, surrogate modeling, kriging interpolation, co-kriging.

## I. Introduction

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 (Fonseca (1995)), which describes the best 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 (Hwang & Masud (1947)).

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 (Miettinen (1999)) (MOO) (also called vector optimization or Pareto optimization) is used to obtain the best possible trade-offs between competing objectives. There are various methods to perform MOO. A widely used approach involves the use of metaheuristic algorithms, such as genetic algorithms (Holland (1975)) (GAs), multi-objective evolutionary algorithm (Zitzler (1999)) (MOEAs), and particle swarm optimization (Eberhart & Shi (1998)) (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 uncommon). Consequently, utilization of metaheuristic algorithms is 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.

The overall objective of this research work is to investigate strategies to accelerate PDE-constrained MOO to enable fast iterative system design. 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.

To limit the scope, the work presented in the paper is focused on the design of aerodynamic surfaces. 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). The following are just a few examples from the literature where MOO is used in aerodynamic design. Guo (2007) 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. Li & Chang (2014) 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 et al. (2014) 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. Kenway et al. (2014) uses the weighted sum method and a gradient-based search algorithm to perform an multi-objective aero-structural optimization. Alfayyadh et al. (2014) 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. Omara & Abidob (2010) uses the strength Pareto evolutionary algorithm (SPEA) for the design of an integrated fuzzy guidance law. Mukesha et al. (2012) uses PSO and GA to solve an aerodynamic shape optimization problem concerning NACA 0012 airfoil using 12 design variables. The 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 as MOO tools. Moreover, in these works, the

high-fidelity model is utilized directly in the MOO process.

MOO of airfoil shapes is considered in this work. The objective is to generate the trade-offs of the airfoil characteristics, i.e., the best possible trade-offs between the drag and pitching moment coefficients are generated at a fixed lift coefficient. This design problem requires computational fluid dynamics (CFD) simulations. Airfoil shapes are typically described using up to 15 parameters, and, in this work, the shapes are parameterized using eight parameters. Therefore, the MOO design problem considered in this work involves small-scale PDE simulations (on the order of half a million unknowns and taking around 20 minutes to solve on HPC) and a low-dimensional design space (8 parameters). Future work, will consider larger scales and higher dimensional problems.

The MOO approach utilized in this work has been recently proposed by Koziel et al. (2016). The approach integrates fast physics-based surrogate models and design space reduction techniques to approximately identify the Pareto front, and, subsequently, refines it using a limited number of computationally expensive system evaluations. A critical step in the MOO process is to reduce the design space to enable the construction of an accurate approximation models, kriging interpolation (Simpson et al. (2001)) is used in this work with a limited number of fast low-fidelity CFD models. The design space reduction is achieved through single-objective optimization (SOO) of each objective separately. The kriging surrogate is then utilized to generate the initial approximate Pareto front using MOEA, which is computationally efficient since the kriging surrogate is very fast to evaluate. The approximate Pareto front is then refined by constructing a co-kriging model (Simpson et al. (2001)) on top of the initial kriging surrogate with a limited number of high-fidelity PDE simulations. The MOO algorithm is implemented in a computational framework which integrates variable-fidelity CFD models with numerical algorithms.

This work investigates several design strategies to solve the multi-objective airfoil design problem using the MOO computational framework described here above. In particular, the search for the Pareto front, using the MOO algorithm proposed in Koziel et al. (2016), is performed in three ways: (i) within the full initial design space (this strategy neglects the design space reduction step), (ii) same as (i), but using a space-mapping-corrected low-fidelity model (Koziel & Leifsson (2012)), and (iii) within the reduced design space. The strategies are compared with regards to the generated Pareto fronts, and the computational cost. The results of each design strategy are benchmarked against the conventional MOO approach of using SOO with an aggregate objective function.

The paper is organized as follows. Section II described the MOO methodology. The design strategies are outlined in Section III, including a discussion of their advantages and disadvantages. The results of numerical investigations are given in Section IV. The paper is concluded in Section V.

## II. Multi-Objective Optimization Methodology

In this section, the formulation of a multi-objective design problem is stated, and the MOO algorithm considered in this work is described. In particular, each step of the MOO algorithm is described in detail, including the kriging and co-kriging interpolation models, as well as the design space reduction procedure.

### A. Multi-Objective Aerodynamic Design Formulation

Denote  $\mathbf{x}$  as the  $n \times 1$  vector of  $n$  design variables. In this work,  $\mathbf{x}$  contains parameters describing the aerodynamic surface of interest. Let  $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}) \ f_2(\mathbf{x}) \dots f_m(\mathbf{x})]^T$  be a  $m \times 1$  vector of  $m$  high-fidelity PDE simulation responses. Examples of responses include the airfoil section drag coefficient  $f_1(\mathbf{x}) = C_{d,f}(\mathbf{x})$ , pitching moment coefficient  $f_2(\mathbf{x}) = C_{m,f}(\mathbf{x})$ , and the section lift coefficient  $f_3(\mathbf{x}) = C_{l,f}(\mathbf{x})$ . Here, the subscript  $f$  denotes that the attribute is obtained by the high-fidelity PDE simulation. 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, in which case  $F_k(\mathbf{x}) = C_{d,f}(\mathbf{x})$ . Another objective would be to minimize the pitching moment coefficient, in which case  $F_k(\mathbf{x}) = C_{m,f}(\mathbf{x})$ . Typically, an airfoil is designed at a fixed lift coefficient, i.e.,  $C_{l,f}(\mathbf{x}) = \text{constant}$ .

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 (1995)), 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 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  $\mathbf{y} \angle \mathbf{x}$  (Fonseca (1995)). 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, 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.

## B. 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 simulations, which is especially challenging in MOO. Therefore, for the sake of computational efficiency, the MOO procedure (Koziel et al. (2016)) utilized in this work exploits, apart from the original, high-fidelity simulation model  $\mathbf{f}$ , its low-fidelity model counterpart  $\mathbf{c}$ . In this work, the low-fidelity model is based on coarse-discretization CFD simulations (the detailed setup is discussed in Sect. B), 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.

The steps of the MOO algorithm (Koziel et al. (2016)) follow:

1. Setup a physics-based surrogate  $\mathbf{s}_0$ ;
2. Perform design space reduction using  $\mathbf{s}_0$ ;
3. Sample the design space and acquire the surrogate model data with  $\mathbf{s}_0$ ;
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 the 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. In this work, a combination of physics-based and data-driven surrogate models is used. The construction of the data-driven surrogates are described in Sects. II.D and II.F, whereas the setup of the physics-based surrogate models is given in Sect. III.
- **Step 2:** Setting up an accurate data-driven surrogate (Step 4) can be very expensive 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 proposed 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 a limited number of model evaluations. The design space reduction methodology is described in Sect. II.C.

- **Step 3:** Latin Hypercube Sampling (Beachkofski & Grandhi (2002)) (LHS) is used to select the shapes within the reduced design space for the kriging model construction. The LHS sampling is described in Sect. II.D.
- **Step 4:** Using the sampled data from Step 3, a kriging interpolation model is constructed. Sect. II.D 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, that the Pareto set obtained is not accurate to the high-fidelity level as we obtain it using the approximate surrogate which is based on the low-fidelity model. The MOEA used in this work is described in Sect. II.E.
- **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 Sect. II.F.
- **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.

## C. Design Space Reduction

The MOO algorithm utilized 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 time frame, even if the dimensionality of the design space is relatively large (say, more than 10 parameters). Here, similarly as in Koziel et al. (2015), 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

$$\mathbf{x}_c^{*(k)} = \arg \min_{\mathbf{l} \leq \mathbf{x} \leq \mathbf{u}} F_k(\mathbf{s}_0(\mathbf{x})) \quad (1)$$

where  $k = 1, \dots, N_{obj}$ . The boundaries of the reduced design space can be then defined as  $\mathbf{l}^* = \min\{\mathbf{x}_c^{*(1)}, \dots, \mathbf{x}_c^{*(N_{obj})}\}$  and  $\mathbf{u}^* = \max\{\mathbf{x}_c^{*(1)}, \dots, \mathbf{x}_c^{*(N_{obj})}\}$ . The concept of the search space reduction is illustrated in Fig. 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.

#### D. Kriging Surrogate Construction

A kriging interpolation surrogate model is constructed in Step 4 of the MOO algorithm presented in Sect. II.B. An outline of the process of constructing the surrogate is shown in Fig. 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.

##### 1. Design of Experiments: Latin Hypercube Sampling

LHS is used to allocate sample points within the design space. 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 (Beachkofski & Grandhi (2002)). The idea of LHS is to use bins to sample the points along each design variable dimension. For example, as shown in Fig. 3, if 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.

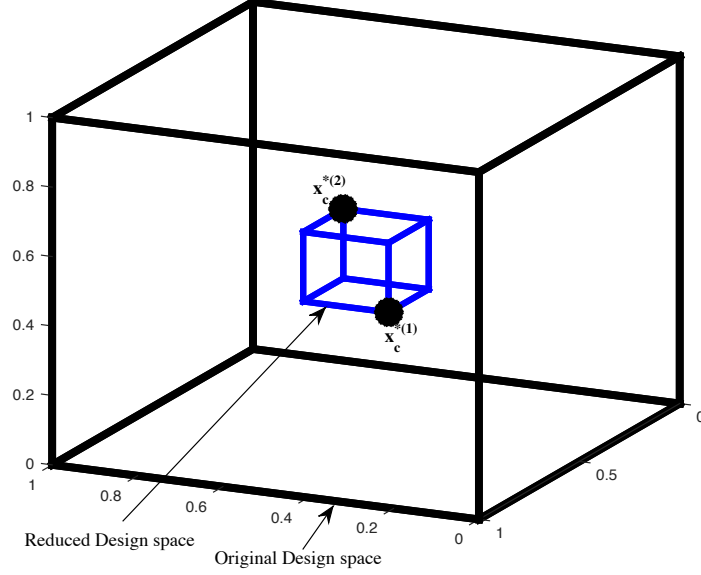


Figure 1: An illustration of the design space reduction technique using single-objective optimization runs (Koziel et al. (2015)). The illustration assumes a three-dimensional design space and two design objectives.

## 2. Data Fitting: Kriging Interpolation

A kriging interpolation model (Simpson et al. (2001)) 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 Sect. II.F. This section briefly provides background information on kriging interpolation. A detailed survey can be found in the literature (Kleijnen (2015), Sacks et al. (1989)).

Let  $X_B = \{\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^N\}$  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 (Simpson et al. (2001)). The regression function  $\mathbf{s}_{KR}(\mathbf{x})$  captures highest variance in the training samples while the GP covers the details related to the interpolation accuracy. This is provided by a kriging interpolant denoted as

$$\mathbf{S}_{KR}(\mathbf{x}) = \mathbf{M}\alpha + \mathbf{r}(\mathbf{x}) \cdot \Psi^{-1} \cdot (\mathbf{f}(X_{B.KR}) - \mathbf{F}\alpha), \quad (2)$$

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  $\alpha$  is a regression function coefficient of the form

$$\alpha = (X'_{B.KR} \Psi^{-1} X_{B.KR})^{-1} X_{B.KR} \Psi^{-1} (\mathbf{f}(X_{B.KR})) \quad (3)$$

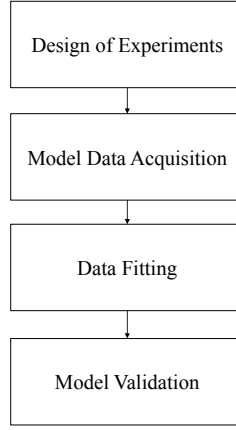


Figure 2: Flowchart describing the data-driven surrogate modelling process

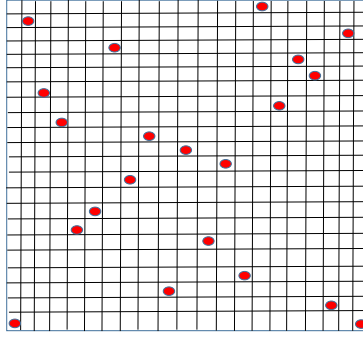


Figure 3: An illustration of Latin Hypercube Sampling.

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

$$\begin{bmatrix} \Psi(\mathbf{x}_{KR}^1, \mathbf{x}_{KR}^1) & \dots & \Psi(\mathbf{x}_{KR}^1, \mathbf{x}_{KR}^{N_{KR}}) \\ \vdots & \ddots & \vdots \\ \Psi(\mathbf{x}_{KR}^{N_{KR}}, \mathbf{x}_{KR}^1) & \dots & \Psi(\mathbf{x}_{KR}^{N_{KR}}, \mathbf{x}_{KR}^{N_{KR}}) \end{bmatrix} \quad (4)$$

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) 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  $\mathbf{x}$  and  $\mathbf{x}'$ ) is defined by

$$\Psi(\mathbf{x}, \mathbf{x}') = \exp \left( \sum_{k=1, \dots, n} -\theta_k |x^k - x'^k|^p \right), \quad (5)$$

where the parameter  $p$  determines the prediction smoothness, while  $\theta_k$ ,  $k = 1, \dots, n$ , denotes the influence sphere of a node on its neighbors in each dimension Koziel et al. (2013). This is helpful for identification of relevant variables as it describes the linearity of the response. Usually,  $p$  is constant while the parameters  $\theta_k$  are determined using Maximum Likelihood Estimation (MLE) (Morris et al. (1993)), where the negative concentrated log-likelihood is minimized using

$$\ln(L) \approx -N_{KR}/2 \times \ln(\sigma^2) - 1/2 \ln(|\Psi|), \quad (6)$$

and

$$\sigma^2 = (\mathbf{f}(X_{B.KR}) - \mathbf{F}\alpha)' \Psi^{-1} (\mathbf{f}(X_{B.KR}) - \mathbf{F}\alpha) / N_{KR}. \quad (7)$$

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.,  $\mathbf{F} = [1 \ 1 \ \dots 1]^T$  and  $\mathbf{M} = (1)$ .

## E. Multi-objective Evolutionary Algorithm

This work utilizes a standard MOEA with fitness sharing, Pareto-dominance tournament selection, and mating restrictions (Fonseca (1995)). 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.

## F. Co-kriging Surrogate Construction

Co-kriging is an extension of kriging which exploits correlations between the models of various fidelities (Simpson et al. (2001)). 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 Forrester et al. (2007) is adopted.

The generation of the co-kriging model is carried out through a sequential construction of two kriging models: the first model  $\mathbf{s}_{KRc}$  derived 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  $s_d = \mathbf{f}(X_{B.KRf}) - \rho \mathbf{c}(X_{B.KRf})$ . The parameter  $\rho$  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  $s_d$ , respectively. Both models use (5) as a correlation function together with constant regression function  $\mathbf{F} = [1 \ 1 \ \dots \ 1]^T$  and  $\mathbf{M} = (1)$ . The final co-kriging model  $\mathbf{s}_{CO}(x)$  is defined similarly as in (2), i.e.,

$$\mathbf{s}_{CO}(\mathbf{x}) = \mathbf{M}\alpha + \mathbf{r}(\mathbf{x}) \cdot \Psi^{-1} \cdot (\mathbf{S}_d - \mathbf{F}\alpha), \quad (8)$$

where the block matrices  $\mathbf{M}$ ,  $\mathbf{F}$ ,  $\mathbf{r}(\mathbf{x})$ , and  $\Psi$  of (6) can be written as a function of the two underlying kriging models  $\mathbf{s}_{KRc}$  and  $\mathbf{s}_{KRd}$  as

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

$$\Psi = \begin{bmatrix} \sigma_c^2 \Psi_c & \rho \cdot \sigma_c^2 \cdot \Psi_c(X_{B.KRc}, X_{B.KRf}) \\ 0 & \rho^2 \cdot \sigma_c^2 \cdot \Psi_c(X_{B.KRf}, X_{B.KRf}) + \sigma_d^2 \cdot \Psi_d \end{bmatrix},$$

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

and

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

### III. Design Strategies

Three design strategies are proposed. The first two search for the Pareto front in the full design space using different types of physics-based surrogate models, whereas the third one searches for the Pareto front in a reduced design space. The details of each strategy is given below, along with a discussion of their characteristics.

#### A. Strategy 1: Full Design Space Search using a Low-fidelity Model

In Strategy 1, each step in the MOO algorithm in Sect. II.B is followed, except that Step 2, the design space reduction step, is not performed. Therefore, the MOO is performed on the full design space. Furthermore, the initial search (Steps 1-5 in the MOO algorithm) is performed using a physics-based surrogate model which is based entirely on a fast low-fidelity model (no correction applied).

More specifically, the physics-based surrogate model utilized in Step 1 is set as  $\mathbf{s}_0(\mathbf{x}) = \mathbf{c}(\mathbf{x})$ , where  $\mathbf{c}(\mathbf{x}) = [c_1(\mathbf{x}) \ c_2(\mathbf{x}) \ \dots \ c_m(\mathbf{x})]^T$  is the response of a model that is of lower fidelity than the high-fidelity PDE simulation model  $\mathbf{f}(\mathbf{x})$ , but much faster to execute. In this work, the responses are the airfoil section drag coefficient  $c_1(\mathbf{x}) = C_{d.c}(\mathbf{x})$ , pitching moment coefficient  $c_2(\mathbf{x}) = C_{m.c}(\mathbf{x})$ , and the section lift coefficient  $c_3(\mathbf{x}) = C_{l.c}(\mathbf{x})$ , where the subscript  $c$  denotes that the attribute is obtained by the low-fidelity model. The low-fidelity model  $\mathbf{c}(\mathbf{x})$  can be developed using (i) models that are based on simplified physics, (ii) the same physics model as the high-fidelity one but with a coarser computational discretization, or (iii) a combination of the two aforementioned approaches. Alternative (ii) is utilized in the numerical investigations. The details are given in Sect. IV.

#### B. Strategy 2: Full Design Space Search using a Space Mapping Surrogate

Strategy 2 follows the same approach as Strategy 1, but with the physics-based surrogate model set up using a response correction technique. In this case, the surrogate model  $\mathbf{s}_0(\mathbf{x})$  is a combination of the low-fidelity model  $\mathbf{c}(\mathbf{x})$ , and a correction applied to its output. In particular, the multi-point output space mapping (OSM) technique (Koziel et al. (2008), Koziel & Leifsson (2012)) is utilized here for this purpose. The correction is based on information from the high-fidelity PDE simulation model  $\mathbf{f}(\mathbf{x})$ .

Multi-point OSM uses correction terms that are directly applied to the response components  $C_{d.c}(\mathbf{x})$ ,  $C_{m.c}(\mathbf{x})$ , and  $C_{l.c}(\mathbf{x})$  of the low-fidelity model. The physics-based surrogate model is defined as (Koziel & Leifsson (2012))

$$\mathbf{s}_0(\mathbf{x}) = \mathbf{A}(\mathbf{x}) \circ \mathbf{c}(\mathbf{x}) + \mathbf{D}(\mathbf{x}), \quad (9)$$

where  $\circ$  denotes component-wise multiplication, and  $\mathbf{A}(\mathbf{x})$  and  $\mathbf{D}(\mathbf{x})$  are multiplicative and additive correction terms, respectively. The response correction parameters  $\mathbf{A}(\mathbf{x})$  and  $\mathbf{D}(\mathbf{x})$  are obtained by solving

$$[\mathbf{A}, \mathbf{D}] = \arg \min_{[\bar{\mathbf{A}}, \bar{\mathbf{D}}]} \sum_{k=1}^N \|\mathbf{f}(\mathbf{x}^k) - (\bar{\mathbf{A}}(\mathbf{x}^k) \circ \mathbf{c}(\mathbf{x}^k) + \bar{\mathbf{D}}(\mathbf{x}^k))\|^2 \quad (10)$$

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

Fortunately, it is not necessary to solve the nonlinear minimization problem (10) directly. By defining the response correction terms as linear functions of the design variables  $\mathbf{x}$  it is possible to calculate them using a least-square optimal solution to a linear regression problem which is equivalent to (10) (Koziel et al. (2008), Koziel & Leifsson (2012)). In particular, the terms are defined as

$$\mathbf{A}(\mathbf{x}) = \begin{bmatrix} a_{0.1} \\ \vdots \\ a_{0.m} \end{bmatrix} + \mathbf{A}_q^T (\mathbf{x} - \mathbf{x}^0), \quad (11)$$

and

$$\mathbf{D}(\mathbf{x}) = \begin{bmatrix} d_{0.1} \\ \vdots \\ d_{0.m} \end{bmatrix} + \mathbf{D}_q^T (\mathbf{x} - \mathbf{x}^0), \quad (12)$$

where  $m$  is the number of model response components,  $\mathbf{A}_q = [a_{ij}]_{i=1, \dots, n, j=1, \dots, m}$ ,  $\mathbf{D}_q = [d_{ij}]_{i=1, \dots, n, j=1, \dots, m}$ , and  $\mathbf{x}^0$  is the center of the design space. The terms  $\mathbf{A}(\mathbf{x})$  and  $\mathbf{D}(\mathbf{x})$  are obtained analytically as least-square optimal solutions to the regression problems

$$\mathbf{A}(\mathbf{x}^k) \circ \mathbf{c}(\mathbf{x}^k) + \mathbf{D}(\mathbf{x}^k) = \mathbf{f}(\mathbf{x}^k) \quad (13)$$

for  $k = 1, \dots, N$ . Derivation of the optimal solutions are given in Koziel & Leifsson (2012).

The number of training points  $N$  has to be sufficient to ensure that the solution to the regression problem exist. For the linear case of (11) and (12), the minimum number of points is  $2n + 1$ . The training set used in this work consists of (i) factorial design with  $2n + 1$  training points 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 also referred to as the star distribution (Koziel et al. (2011)), (ii) additional 10 points allocated using LHS.



### C. Strategy 3: Reduced Design Space Search using a Low-fidelity Model

In Strategy 3, the full MOO algorithm in Sect. II.B is executed. As in Strategy 1, the physics-based surrogate model utilized in Step 1 is set as  $\mathbf{s}_0(\mathbf{x}) = \mathbf{c}(\mathbf{x})$ .

### D. Discussion

Table 1 gives a comparison of the main differences of the three design strategies. Design strategies 1 and 2 search the full design space. This is accomplished by not performing Step 2 of the MOO algorithm in Sect. II.B. Design strategy 1 utilizes a low-fidelity model to construct the kriging model which is used to approximate the Pareto front. The advantage is that the low-fidelity model is fast, and, therefore, the approximate Pareto front can be found quickly. In this case, the conjecture is that the global refinement step using the co-kriging surrogate will account for any discrepancies between the low- and high-fidelity models. A possible disadvantage is that several iterations are required for the refinement step. Design strategy 2 uses a corrected low-fidelity model to construct the kriging model, and further correction is achieved in the refinement step. The conjecture in this case is that although there is additional cost to setup the initial kriging surrogate, the number of refinement iterations can be reduced. Design strategy 3 searches the reduced design space using a low-fidelity model. It is likely that the reduced design space will be orders of magnitude smaller (volume-wise) than the full design space. Therefore, fewer samples may be needed to setup the data-driven surrogates. Moreover, setting up the initial kriging model using the low-fidelity model should be sufficient since the refinement step will handle the correction.

## IV. Numerical Investigations

In this section, the proposed design strategies of Sect. III are applied to constrained MOO of airfoil shapes in two-dimensional inviscid transonic flow. The case considers transonic airfoil shapes with eight design variables, and two conflicting objectives. In particular, the best possible trade-offs for the drag and pitching moment coefficients are generated at a constant lift coefficient. The section is organized as follows. The problem and the computational models are described in detail first. The results of numerical investigations

Table 1: Comparison of the setup of the design strategies.

Strategy	Design Space	Physics-based surrogate
1	Full	Low-fidelity model
2	Full	Space-mapping-corrected model
3	Reduced	Low-fidelity model

for each design strategy follow along with their comparison. Strategy 3 is studied in more detail by varying its setup parameters. The section concludes with a comparison of the results of the proposed design strategies with the results of the conventional approach of single-objective optimization of an aggregate objective function.

## A. Problem Description and Model Setup

A description of the problem formulation, the airfoil shape parameterization, training data sampling, variable-fidelity computational fluid dynamics (CFD) modeling follows.

### 1. Formulation of the MOO Problem

Multi-objective airfoil design in transonic flow at fixed lift is considered. In particular, the free-stream Mach number is set to  $M_\infty = 0.734$ , and the lift coefficient is fixed at  $C_{l,f}(\mathbf{x}) = 0.824$ . The first objective is to minimize  $F_1(\mathbf{x}) = C_{d,f}(\mathbf{x})$ , and the second objective is to minimize  $F_2(\mathbf{x}) = C_{m,f}(\mathbf{x})$ . A constraint on the cross-sectional area is imposed, i.e.,  $A(\mathbf{x}) \geq A_{baseline}$ , where  $A(\mathbf{x})$  the cross-sectional area of a given design  $\mathbf{x}$  nondimensionalized with the chord squared, and  $A_{baseline}$  is a baseline reference value.

### 2. Design Space

The airfoil shapes are parameterized using B-spline curves for each surface. The airfoil surfaces are written in parametric form as Farin (1993)

$$x(t) = \sum_{i=1}^{n+1} X_i N_{i,k}(t), \quad z(t) = \sum_{i=1}^{n+1} Z_i N_{i,k}(t), \quad (14)$$

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 (14) is continuous. The control points are used as design variables and allowed only to move freely vertically as shown in Fig. 4. Thus, the design variable vector is  $\mathbf{x} = [Z_1 \ Z_2 \ \dots \ Z_{n+1}]^T$ , and the corresponding  $X_i$  coordinates are held fixed during the optimization process. A total of eight design variables are used in this work with four for each surface (see Fig. 4).

The RAE 2822 airfoil is used as a baseline shape (shown in Fig. 4). By setting the x-locations of design variables as  $\mathbf{X} = [\mathbf{X}_u; \mathbf{X}_l]^T = [0.0 \ 0.15 \ 0.45 \ 0.8; 0.0 \ 0.35 \ 0.6 \ 0.9]^T$ , the RAE 2822 shape is fitted to a B-spline curve with the baseline design variable vector set to  $\mathbf{x}_0 = [\mathbf{x}_u; \mathbf{x}_l]^T = [0.0175 \ 0.04975 \ 0.0688 \ 0.0406; -0.0291 \ -0.0679 \ -0.03842 \ 0.0054]^T$ .

The bounds of the design space are defined by  $\mathbf{l} = (1 - \text{sign}(\mathbf{x}_0) \cdot 0.15) \circ \mathbf{x}_0$  and  $\mathbf{u} = (1 + \text{sign}(\mathbf{x}_0) \cdot 0.15) \circ \mathbf{x}_0$ . Using the definition, the lower and upper bounds on  $\mathbf{x}_0$  are set as  $\mathbf{l} =$

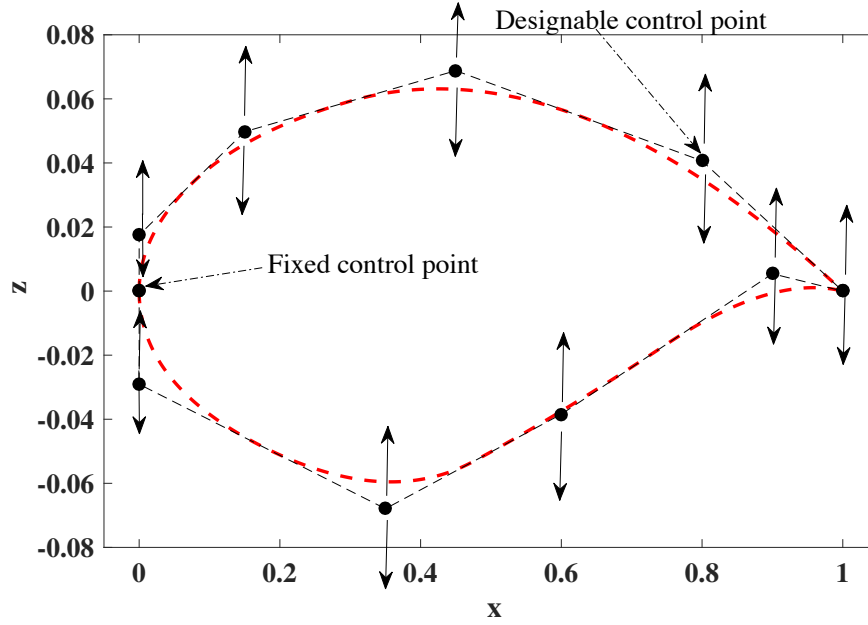


Figure 4: An example B-spline parameterization of an airfoil surface (the RAE 2822 is shown here). The designable control points are restricted to vertical movements only.

$[0.0105 \ 0.0414 \ 0.0537 \ 0.0200; -0.0369 \ -0.0808 \ -0.0666 \ -0.0265]^T$  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  $A_{RAE2822} = 0.0779$ .

### 3. Training Points

The 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 Sect. II.D). 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 results are given in Sect. IV.B). Figure 5 shows the baseline airfoil, as well as a few samples from the base set. Figure 6 shows the design points (in black) for two of the control points.

### 4. Computational Fluid Dynamics Modeling

The Stanford University Unstructured (SU2) computer code is utilized for the fluid flow simulations (Palacios et al. (2013)). 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 (Jameson et al. (1981)). Three multi-grid levels are used for solution acceleration. Asymptotic convergence to a steady state

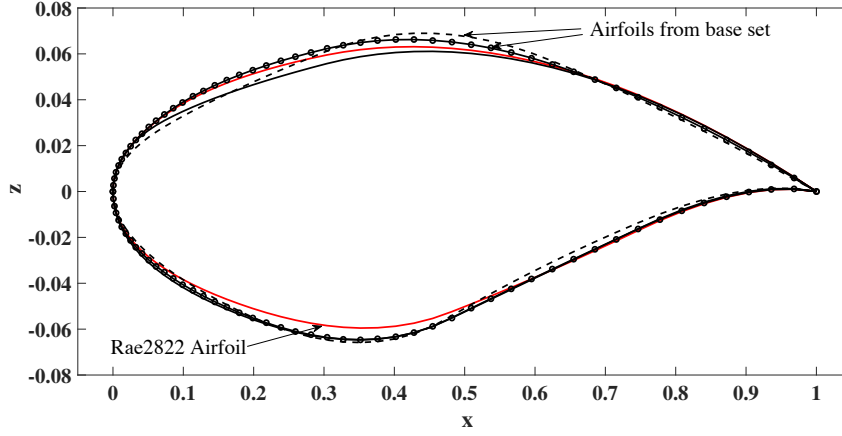


Figure 5: The baseline airfoil (RAE 2822) and sample airfoils from the base training set.

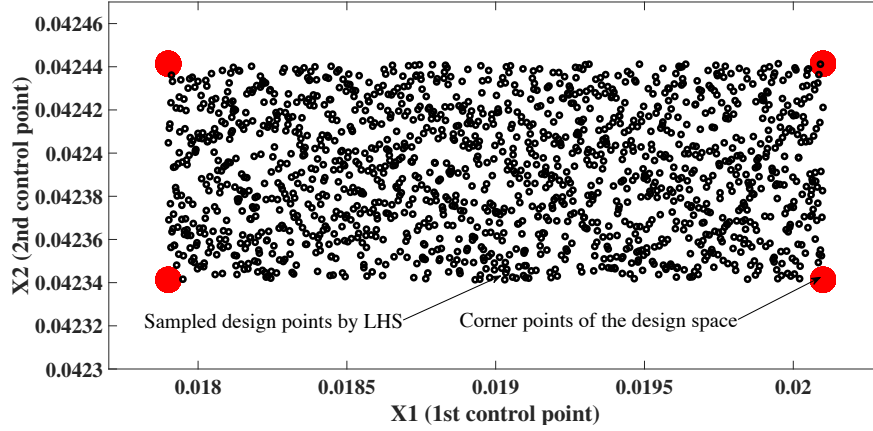


Figure 6: Example training points sampled using Latin Hypercube Sampling.

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 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 2, revealed that the  $512 \times 512$  mesh (shown number 5 in the table) is required for convergence within 0.2 drag counts when compared with the next mesh. The flow simulation for Mesh 5 takes around 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  $\mathbf{f}(\mathbf{x})$ , and Mesh 3 as the low-fidelity model  $\mathbf{c}(\mathbf{x})$ . Figure 7 shows the computational

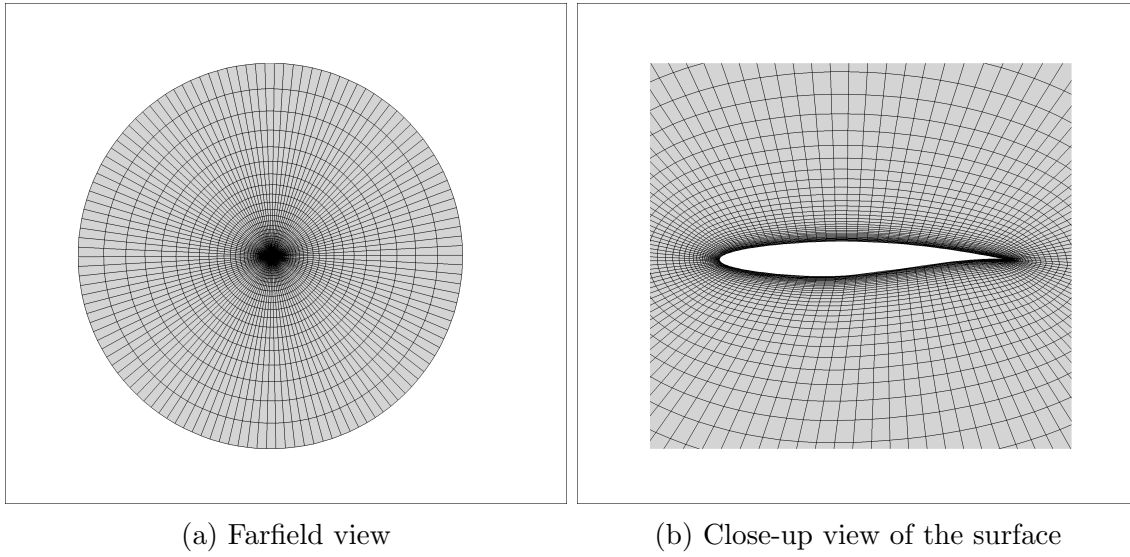


Figure 7: The computational O-mesh.

Table 2: Results of the grid convergence study at  $M_\infty = 0.734$  and  $C_l = 0.824$ .

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

grids. For the low-fidelity model, the maximum number of solver iterations is set to 300. Figure 8a shows an example of the solver convergence of the low-fidelity model. The high-to-low simulation time ratio is on average around 30. A comparison of the pressure distributions of the high- and low-fidelity models, shown in Fig. 8b, 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 2). 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  $\Delta C_m = 0.00127$ .

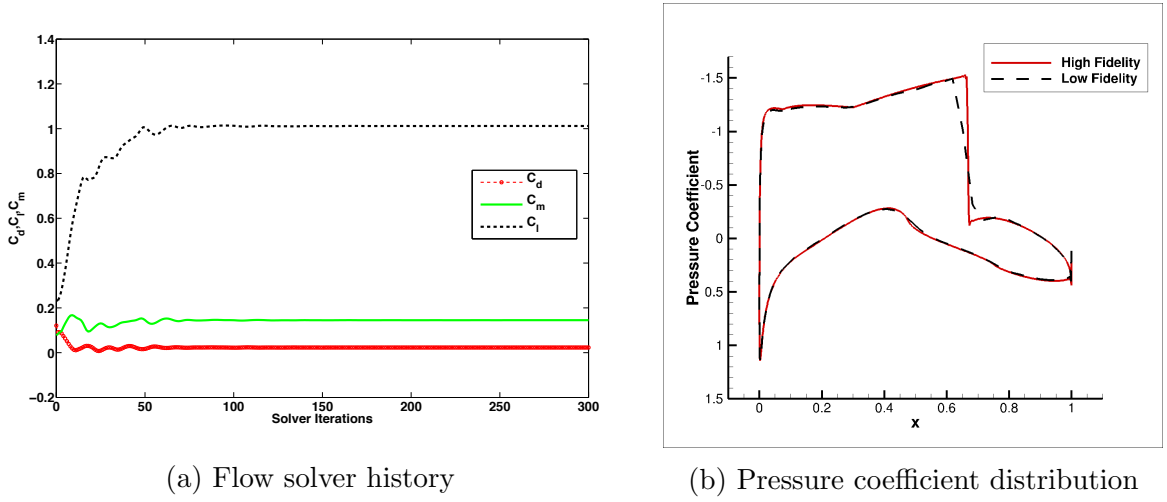


Figure 8: Low-fidelity model ( $64 \times 64$  mesh) output characteristics at  $M_\infty = 0.734$ .

## B. Multi-objective Optimization Results

The results of the numerical investigations are presented in this section. In particular, the results for each design strategy are given separately for clarity, and, subsequently, compared with each other and a benchmarking approach.

### 1. Results of Strategy 1: Full Design Space Search using a Low-fidelity Model

With Design Strategy 1, the MOO algorithm was manually stopped after 7 iterations. Figure 9 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 was repeated 7 times. Figure 10 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.

Table 3 shows how the root mean square error of the kriging model varies with the number of samples. In this case, the minimum number of samples required to have the MOO algorithm (partially) converge was 1,600. Moreover, the minimum number of high-fidelity samples per design iteration was 9. 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 fully converge even after 9 iterations and hence the total time can be up to around 3 days. Assuming that only the high-fidelity model is utilized in the MOO algorithm, and assuming that 1,663 samples are still needed, the total time would be around 34.6 days.

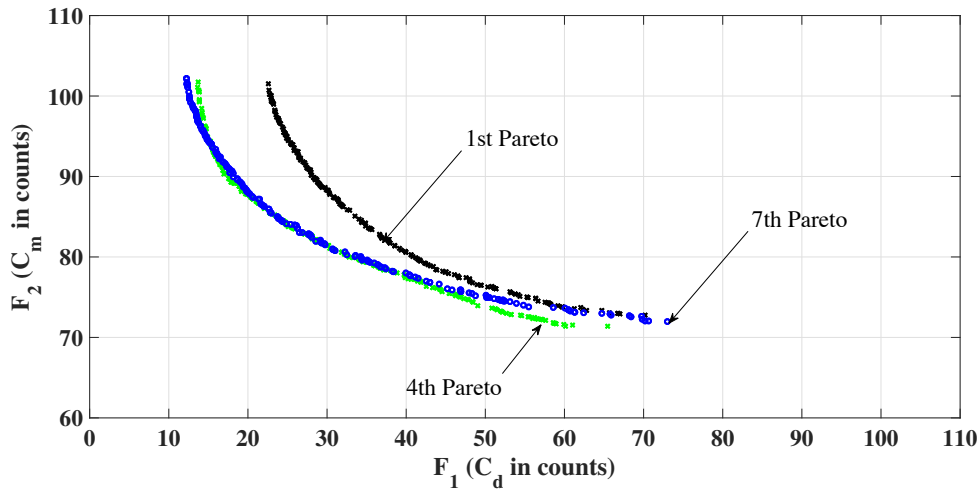


Figure 9: Results of Design Strategy 1 showing the Pareto fronts obtained at a few iterations.

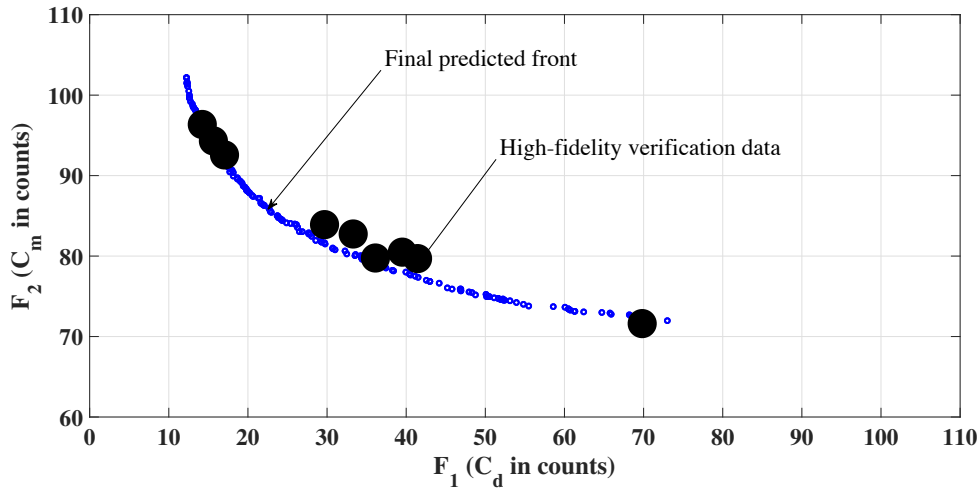


Figure 10: The final Pareto front of Design Strategy 1 with high-fidelity validation samples.



Table 3: Relative root mean square error (RMSE) of the initial kriging model in Step 4 of the MOO algorithm in Design Strategy 1 for different number of samples.

Samples	Relative RMSE (%)
100	6.04
500	2.62
1,000	1.91
1,600	1.45

## 2. Results of Strategy 2: Full Design Space Search using a Space Mapping Surrogate

For Design Strategy 2, 1,600 points are used for the setup of the initial kriging surrogate (the same number as for Design Strategy 1 is used for comparison purposes), as well as extra 30 feasible points are generated 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. While collecting points to calculate the OSM correction parameters, care is taken so 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 (same as for Design Strategy 1), evaluated along the Pareto front to generate the co-kriging model  $\mathbf{s}_{CO}$  until convergence.

Figure 11 shows the Pareto front obtained with Design Strategy 2. In this case, three iterations of the MOO are enough to fully converge. It can be observed from Fig. 11 that the Pareto front predicted in the third iteration is converged within one drag count, i.e, after three iterations, the agreement between the predicted front and the refinement samples is within one drag count.

The overall computational cost of the MOO process for Design Strategy 2 in terms of the number of evaluations are the initial 1,600 low-fidelity model evaluations, and 30 low- and high-fidelity model evaluations for the 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 around 3,340 minutes, or 2.3 days.

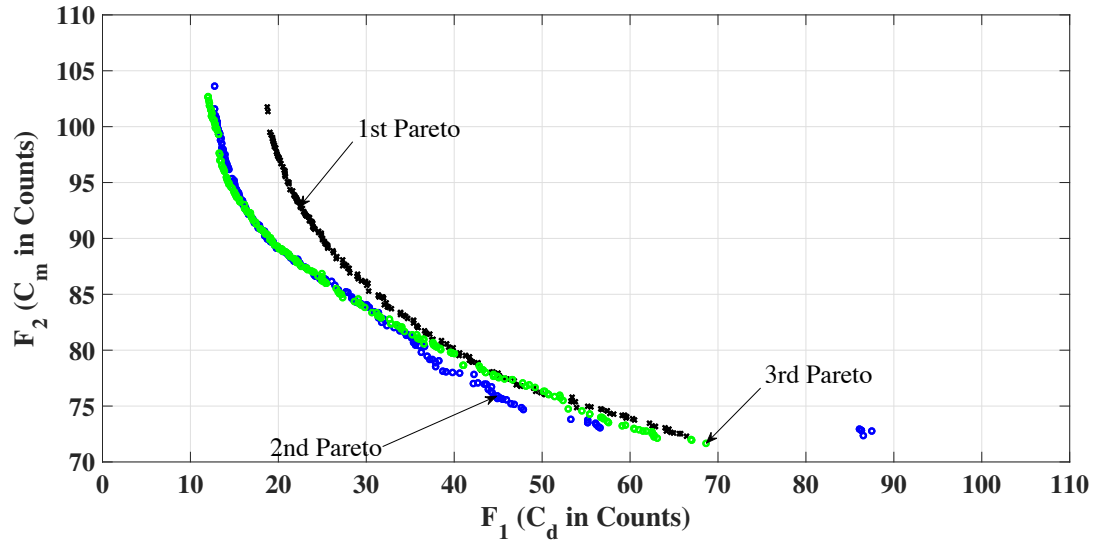


Figure 11: Results of Design Strategy 2 showing the Pareto fronts obtained for each iteration.

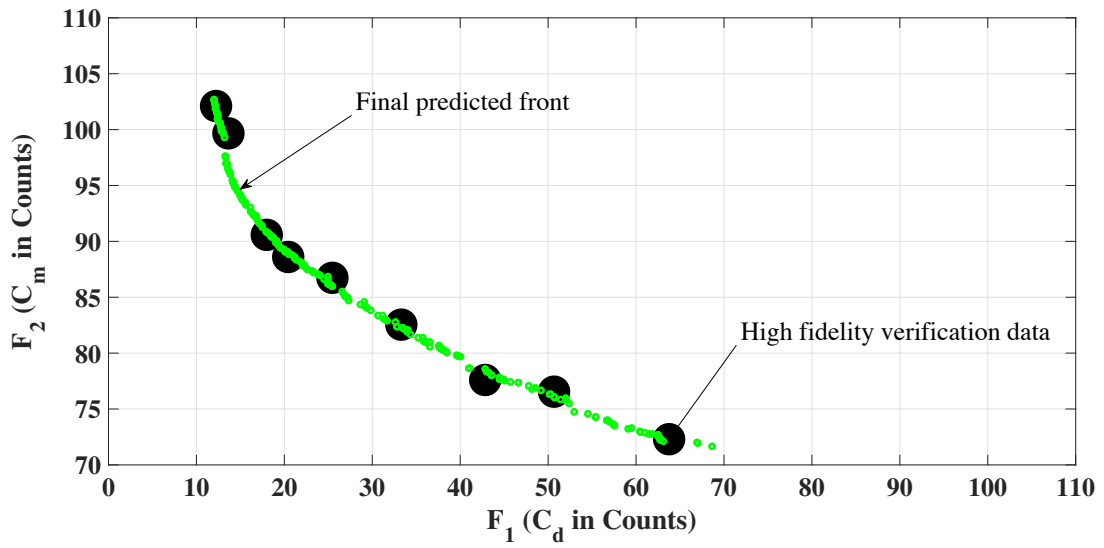


Figure 12: The final Pareto front of Design Strategy 2 with high-fidelity validation samples.

### 3. Results of Strategy 3: Reduced Design Space Search using a Low-fidelity Model

In Design Strategy 3, the design space is reduced by running two single-objective optimization runs to obtain an approximation of the extreme points of the Pareto front. The pattern search method 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:

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

and

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

The pattern search method needed 256 function evaluations in both optimization runs to reach to the minimum, and optimum designs found are:

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

and

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

The reduced design space (according to Sect. II.C) is, therefore, defined by  $\mathbf{l}^* = \min(\mathbf{x}_c^{*(1)}, \mathbf{x}_c^{*(2)})$ , and  $\mathbf{u}^* = \max(\mathbf{x}_c^{*(1)}, \mathbf{x}_c^{*(2)})$ . Comparing  $\mathbf{x}_c^{*(1)}$  and  $\mathbf{x}_c^{*(2)}$ , it is found that the 5th design is same in  $\mathbf{l}^*$  and  $\mathbf{u}^*$ , which means that the value of that design variable is constant at -0.02938, and, hence, the dimension of the design space can be reduced by removing that design variable. Now the reduced design space has seven design variables instead of eight. Thus, the bounds of the reduced design space are:

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

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 (same as for Design Strategies 1 and 2), 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 (same as Design Strategies 1 and 2) which are sampled uniformly along the Pareto front predicted by the surrogate model. The MOO process converged in two iterations. Figure 13 shows the Pareto fronts obtained in the design iterations. In Fig. 14 it can be observed that there is a good agreement between the predicted front and the validation samples after two iterations, and the algorithm has reached the convergence (within the prescribed one drag count). In this case, 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.

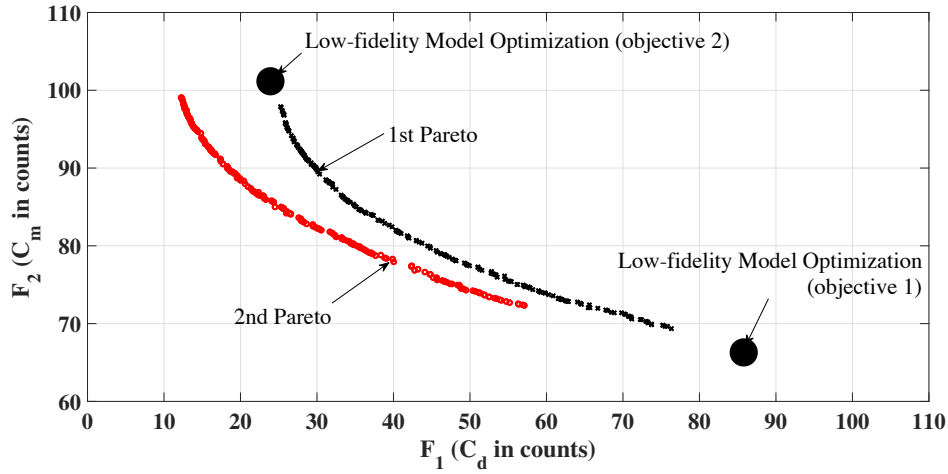


Figure 13: Results of Design Strategy 3 showing the Pareto fronts obtained at each iteration. The approximate extreme points obtained by the single-objective optimization runs are also shown.

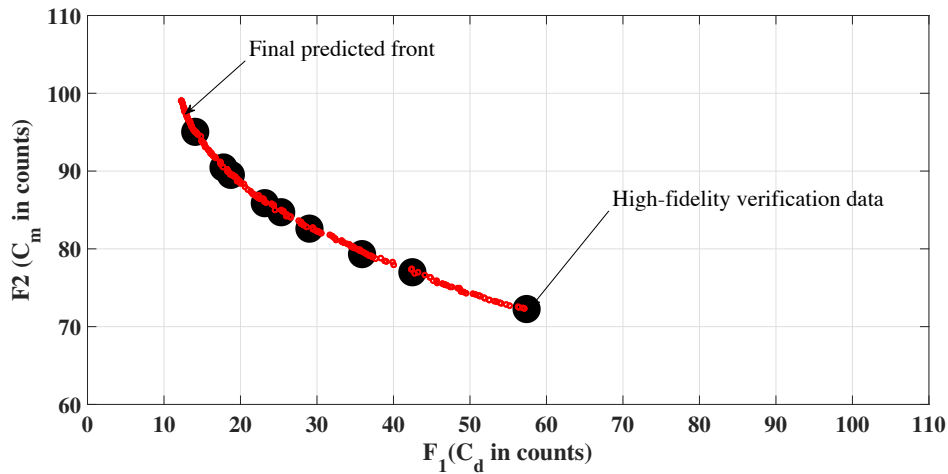


Figure 14: The final Pareto front of Design Strategy 3 along with high-fidelity validation samples.

#### 4. Comparison of the Design Strategies

Figure 15 compares the final Pareto fronts by the three design strategies. It can be observed that the fronts are comparable in shape. Table 4 summarizes the computational cost. Design Strategy 1 was the least efficient as it needed 3 days and was not fully converged after 7 iterations. Design Strategy 2 gave good results and converged within 3 iterations, or 2.3 days. Design Strategy 3 was the most efficient approach as it converged within 2 iterations, or 1.8 days.

Three designs were selected along the final Pareto front of Design Strategy 3 (as shown in Fig. 16) and their characteristic features are given in Figs. 17, 18, 19, 20 and 21.

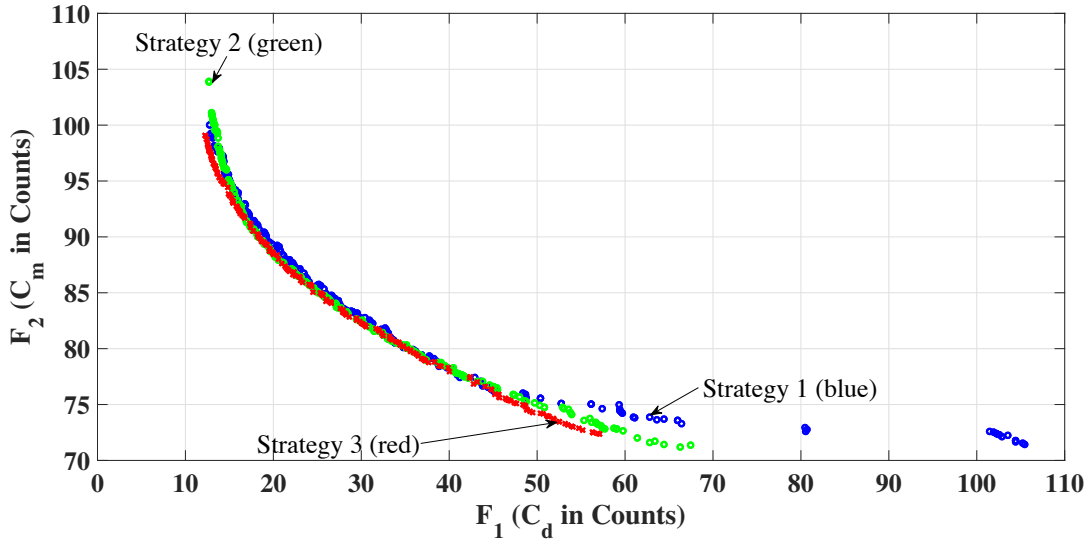


Figure 15: Comparison of the final Pareto fronts obtained by the three strategies.

Table 4: Comparison of the computational 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

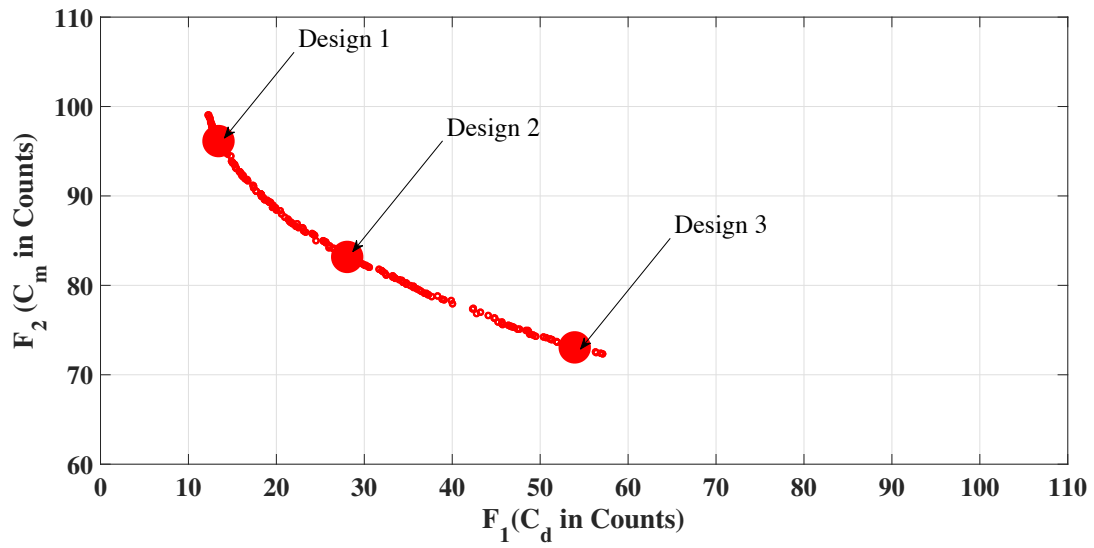


Figure 16: Designs selected along the final Pareto front of Design Strategy 3 for visualization.

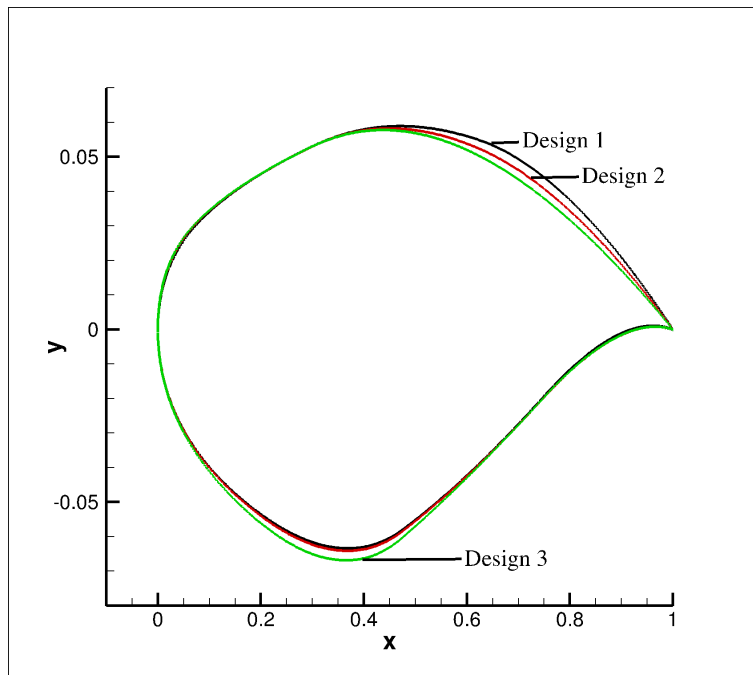


Figure 17: Airfoil shape of the designs selected from the final Pareto front of Design Strategy 3.

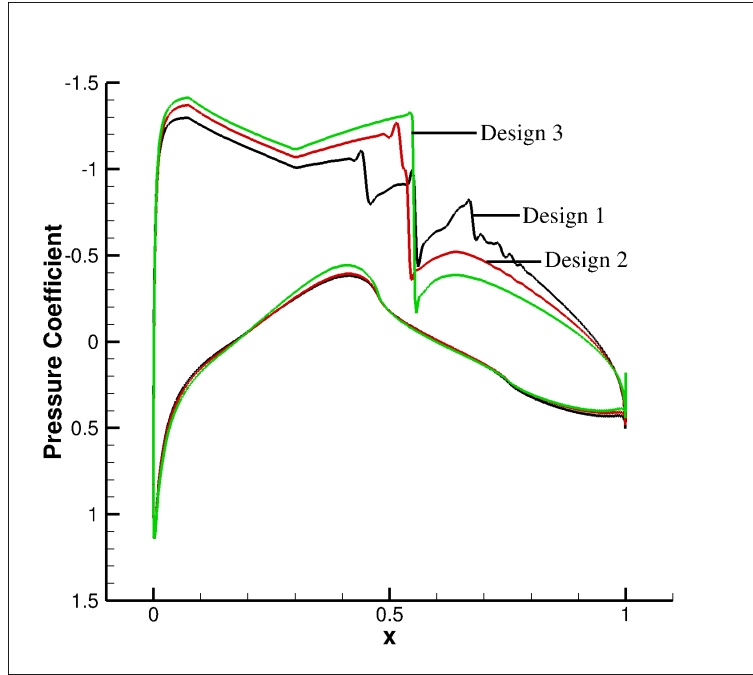


Figure 18: Pressure coefficient of designs selected from the final Pareto front of Design Strategy 3.

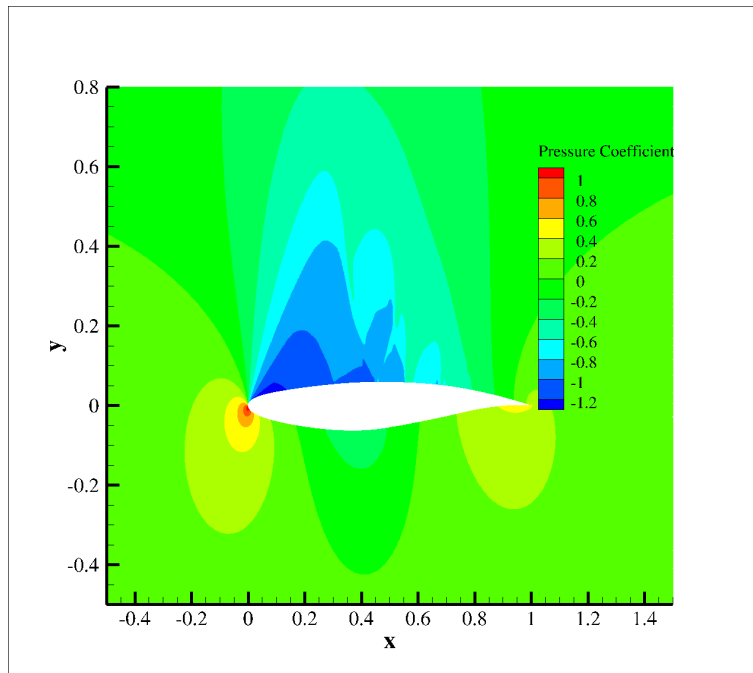


Figure 19: Pressure coefficient contours for design 1.



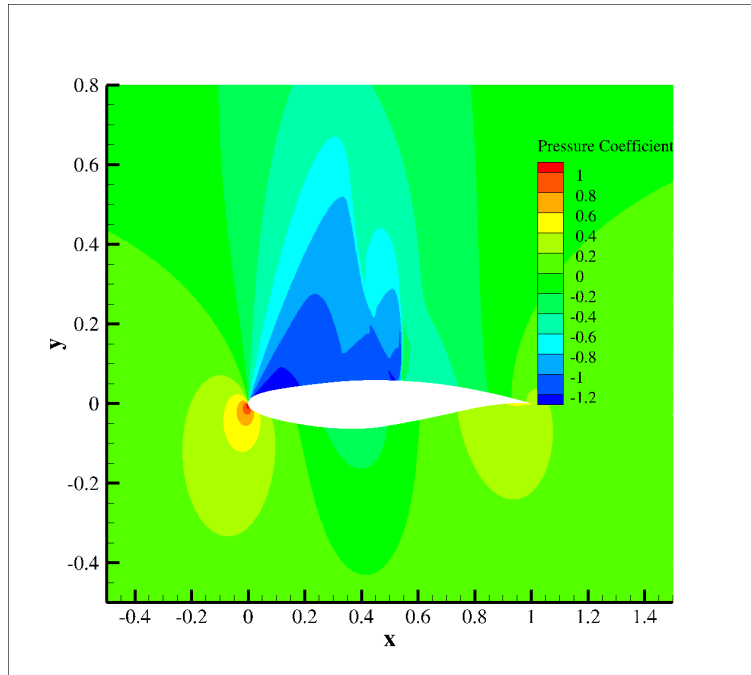


Figure 20: Pressure coefficient contours for design 2.

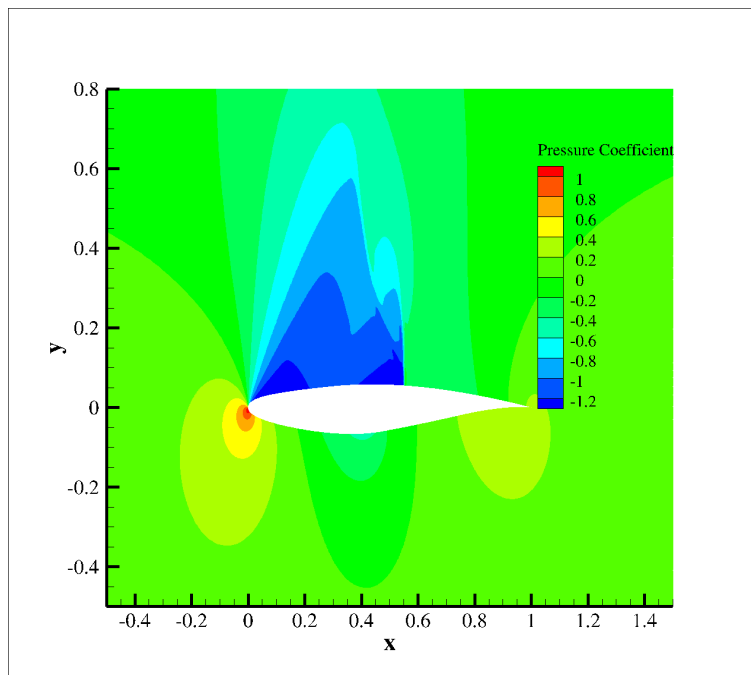


Figure 21: Pressure coefficient contours for design 3.

### 5. Parametric Study of Design Strategy 3

A parametric study of Design Strategy 3 is performed by varying the number of low-fidelity model sampling points. For each case, the MOO algorithm executed. Initial runs used 1,600 low-fidelity designs. Here, the sampling designs are 500, 300, 100, and 50. The RMSE values for Design Strategy 3 is given in Table 5. It can be seen that the RMSE is very low for the 1,600 samples (or 0.17%), indicating that the sampling number is too high. At 50 samples, the RMSE is 3.4%. At 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 Design Strategy 1) is around 1.45%. This fact, and that Design Strategies 1 and 3 yield similar results, indicate that a relative RSME below approximately 1.5% is adequate in this particular test case.

The results of the MOO runs with kriging models generated by 500, 300, and 100 low-fidelity model samples are given in Figs. 22, 23, and 24, 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, 3 refinement points are used (using less than that was unsuccessful). 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 25 shows the results. Again, the MOO algorithm converges within 2 iterations and compares well with the validation samples.

Figure 26 compares the 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 6 give a comparison with the original results.

Table 5: 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

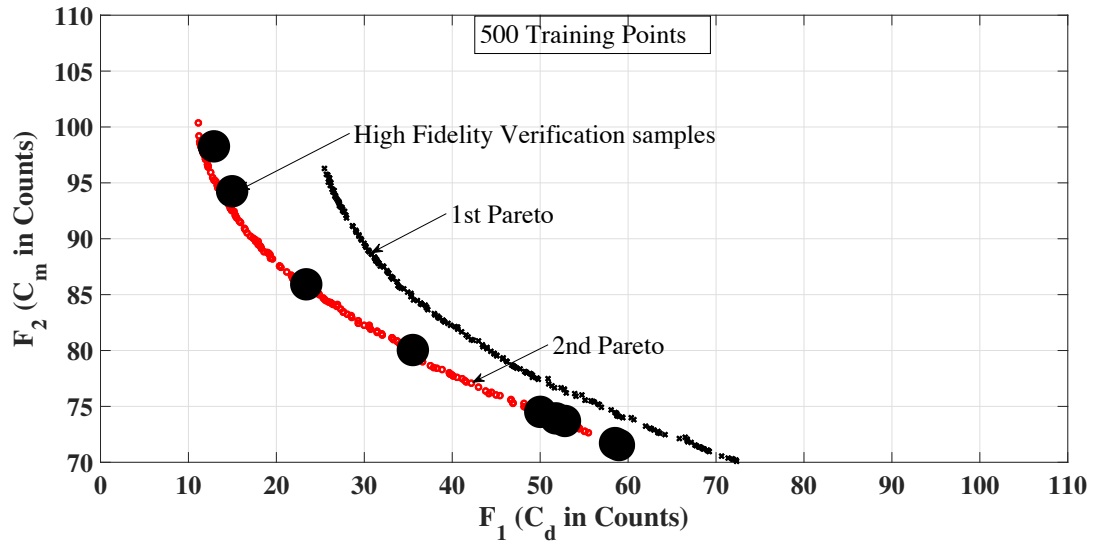


Figure 22: Results of Design Strategy 3 with 500 initial sampling points.

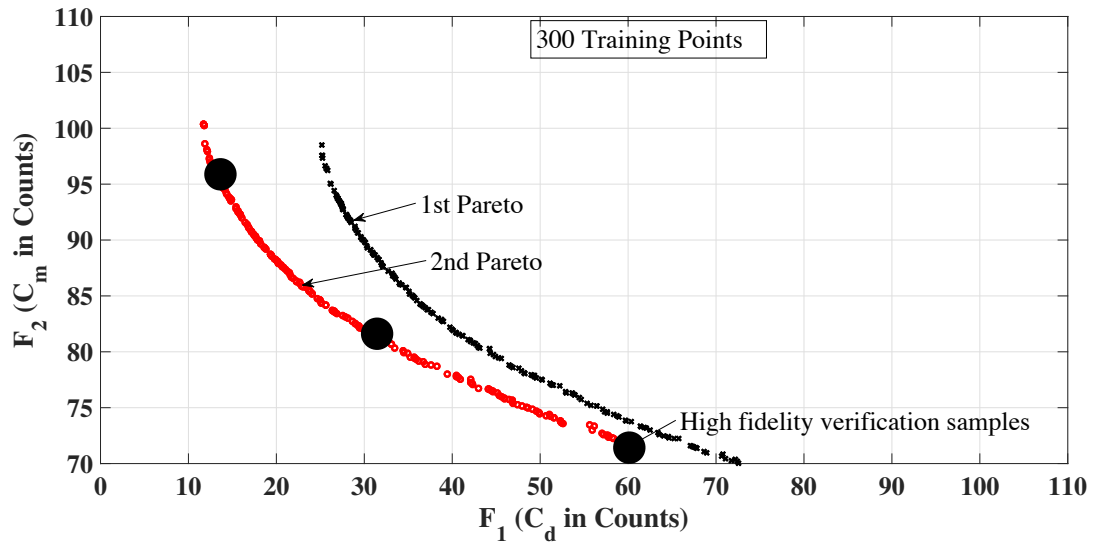


Figure 23: Results of Design Strategy 3 with 300 initial sampling points.

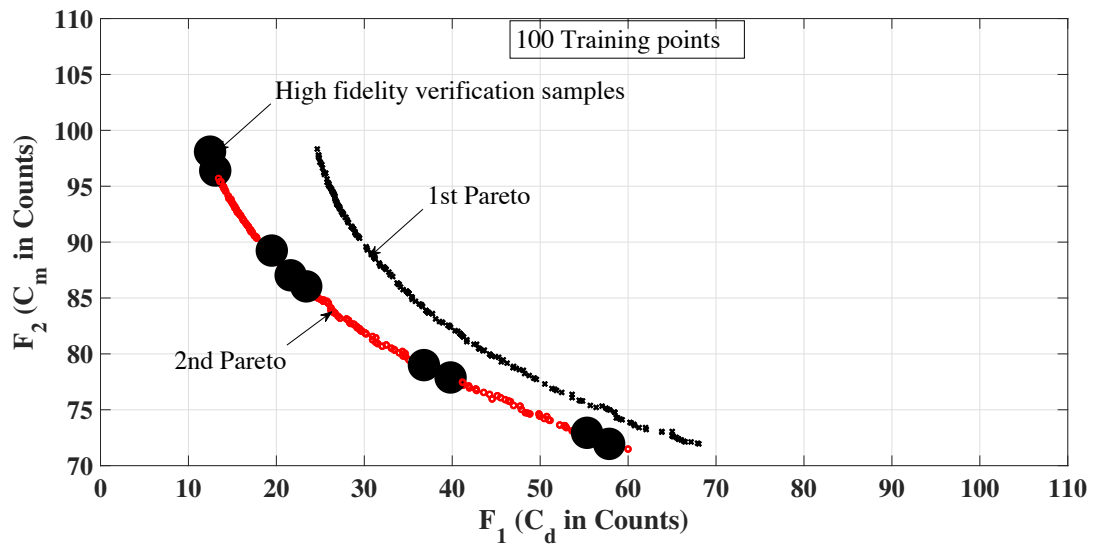


Figure 24: Results of Design Strategy 3 with 100 initial sampling points.

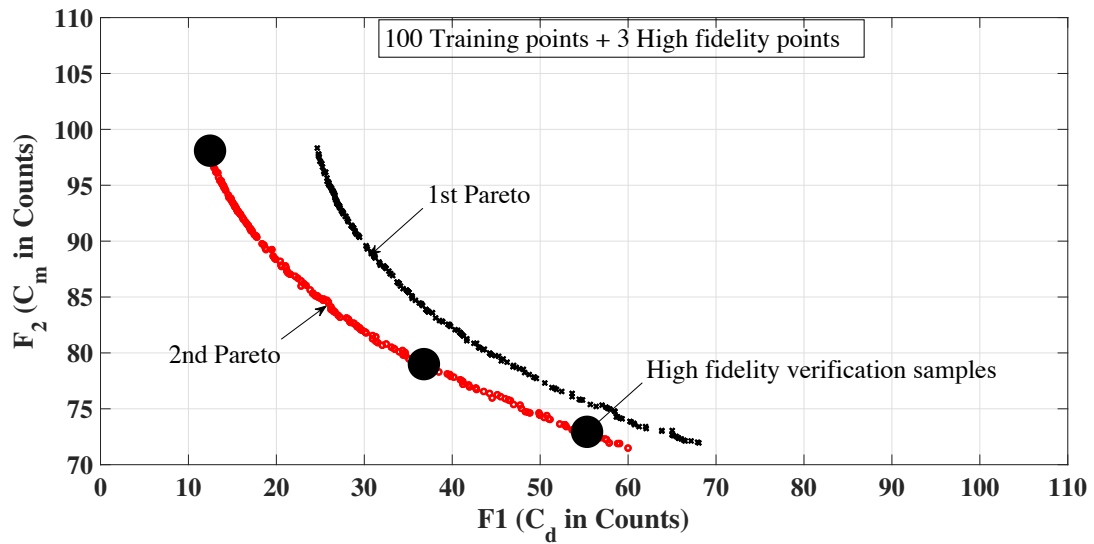


Figure 25: Results of Design Strategy 3 with 100 initial sampling points and 3 refinement points.

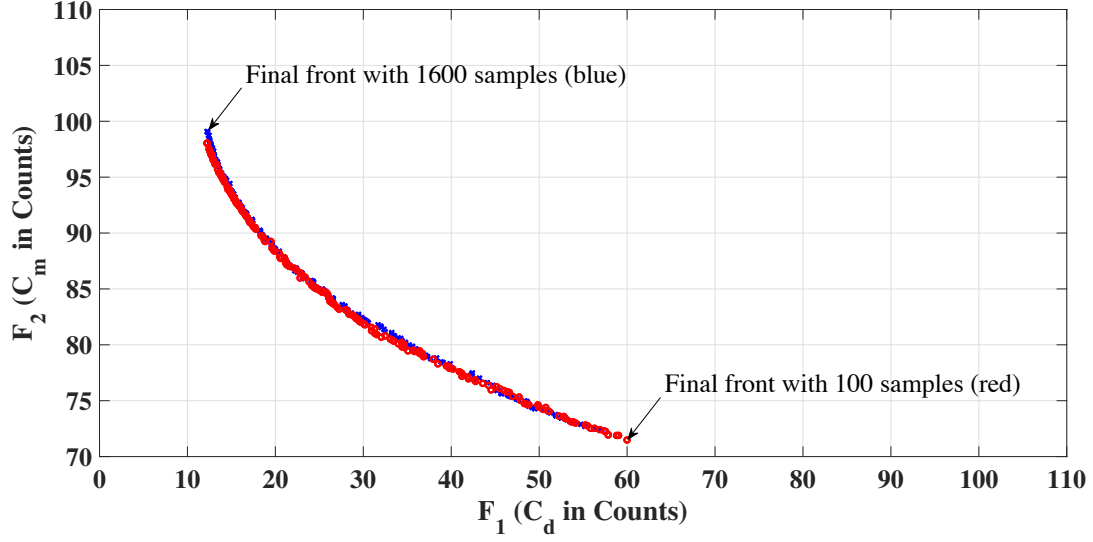


Figure 26: Results of Design Strategy 3 showing the Pareto fronts obtained with 1,600 and 100 initial sampling points.

Table 6: Comparison of the computational cost of the original three strategies, with the refined Design Strategy 3 (indicated by 3\*).

Strategy	Iterations	N <sub>c</sub>	N <sub>f</sub>	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

#### 6. Comparison with the Weighted-sum Approach

The results of the proposed design strategies are compared with the results obtained by single-objective optimization of an aggregate objective function, a so-called weighted-sum approach (see, e.g., Fonseca (1995)). Four points on the Pareto front are found using this approach. In particular, the two extreme points, and two other points on the front.

The extreme points are found as follows:

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

and

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

The results are:

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

and

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

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:

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

Figure 27 shows a comparison of the Pareto fronts obtained by the proposed design strategies and the weighted-sum approach. It can be seen that the final Pareto fronts compare well with each approach.

## V. Conclusion

Design strategies for accelerated multi-objective optimization (MOO) of aerodynamic surfaces using high-fidelity simulations have been presented. The strategies exploit physics-based models, kriging and co-kriging interpolation models to construct accurate and fast surrogates further utilized to search the original or reduced design spaces. Numerical investigations are performed using a problem of transonic airfoil shape. It is found that searching the reduced design space produces the same Pareto front as when searching the entire space. Furthermore, searching the reduced design space requires an order of magnitude fewer samples. Thus, significant computational savings are achieved.

The particular design problem investigated in this work is characterized by continuous and smooth Pareto fronts. In this case, the proposed design strategies and MOO framework

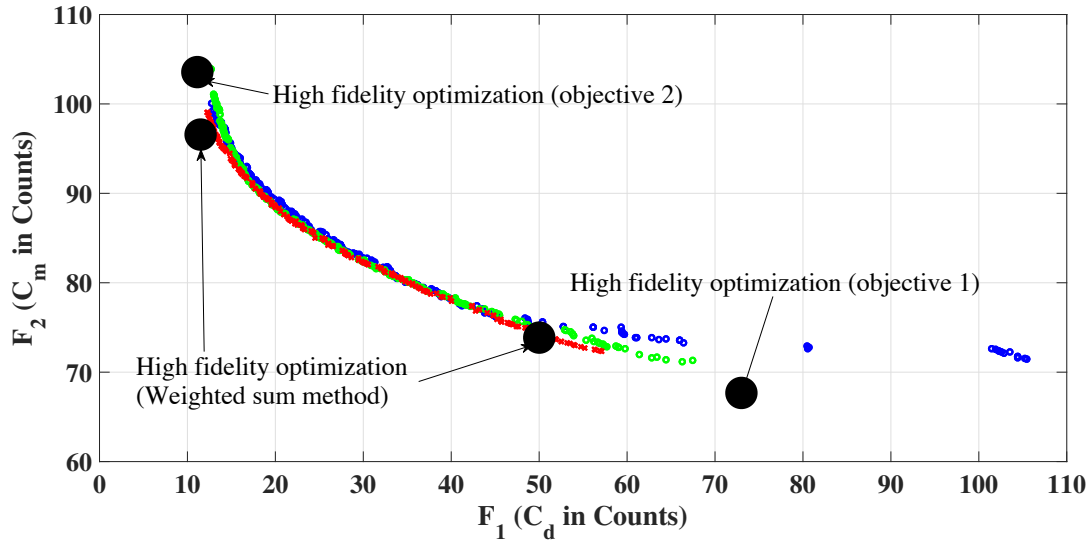


Figure 27: A comparison of the Pareto fronts obtained by the proposed design strategies with the weighted-sum approach (a single-objective optimization of an aggregate objective).

work well. Consequently, the number of samples can be reduced significantly. Future work will be focused on design problems where the Pareto fronts may be less tractable, in particular, consist of several disjoint regions and with noisy objectives. It is expected that this will lead to increased number of the training sets and additional computational overhead. In conjunction with that investigation, the variation of the computational cost with the complexity of the problem, in terms of the number of design variables and objectives, will be studied.

## References

- Alfayyadh, E., Bakhy, S. & Shkara, Y. (2014), '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, July 25-27* **2**.
- Beachkofski, B. & Grandhi, R. (2002), '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*.
- Eberhart, R. & Shi, Y. (1998), 'Comparison between genetic algorithms and particle swarm optimization', *Evolutionary Programming VII, Springer Berlin, Germany* pp. 611–616.
- Farin, G. (1993), *Curves and Surfaces for Computer Aided Geometric Design*, Academic



Press, Boston, MA.

- Fonseca, C. (1995), Multiobjective genetic algorithms with applications to control engineering problems, PhD thesis, Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, UK.
- Forrester, A., Sobester, A. & Keane, A. (2007), ‘Multi-fidelity optimization via surrogate modelling’, *Royal Society, Proceedings of the royal Society* **463**(2088), 3251–3269.
- Guo, S. (2007), ‘Aeroelastic optimization of an aerobatic aircraft wing structure’, *Aerospace Science and Technology* **11**(5), 396–404.
- Holland, J. (1975), *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Ann Harbour, Michigan.
- Hwang, C. & Masud, A. (1947), *Multiple objective decision making, methods and applications: a state-of-the-art survey*, Springer-Verlag, Berlin, Germany.
- Jameson, A., Schmidt, W. & Turkel, E. (1981), ‘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*.
- Kai, A., Kennedy, G. & Martins, J. (2014), ‘Concurrent aerostructural topology optimization of a wing box’, *Computers and Structures* **134**, 1–17.
- Kenway, G., Martins, J. & Kennedy, G. (2014), ‘Aerostructural optimization of the common research model configuration’, *15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, June 16–20*.
- Kleijnen, J. (2015), *Design and Analysis of Simulation Experiments*, Springer International Publishing, Switzerland.
- Koziel, S., Bekasiewicz, A. & Leifsson, L. (2015), ‘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*.
- Koziel, S., Cheng, Q. & Bandler, J. (2008), ‘Space mapping’, *IEEE Microwave Magazine* **9**(6), 105–122.
- Koziel, S., Echeverra-Ciaurri, D. & Leifsson, L. (2011), *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.

- Koziel, S. & Leifsson, L. (2012), ‘Knowledge-based airfoil shape optimization using space mapping’, *AIAA Paper 2012-3016, 30th AIAA Applied Aerodynamics Conference, New Orleans, Louisiana, June 25-28* .
- Koziel, S., Ogurtsov, S. and Couckuyt, I. & Dhaene, T. (2013), ‘Variable-fidelity electromagnetic simulations and co-kriging for accurate modeling of antennas’, *IEEE Trans. Antennas Prop.* **61**(3), 1301–1308.
- Koziel, S., Tesfahunegn, Y. & Leifsson, L. (2016), ‘Expedited constrained multi-objective aerodynamic shape optimization by means of physics-based surrogates’, *Applied Mathematical Modelling* **40**(15-16), 7204–7215.
- Li, W. & Chang, P. (2014), ‘Aeroelastic optimization study based on the x-56a model’, *AIAA Aviation, Atlanta, GA, June 16–20* .
- Miettinen, K. (1999), *Nonlinear Multiobjective Optimization*, Kluwer Academic Publishers, Boston, MA.
- Morris, M., Mitchell, T. & Ylvisaker, D. (1993), ‘Design and analysis of computer experiments: use of derivatives in surface prediction’, *Technometrics* **35**(3), 243–255.
- Mukesh, R., Pandiyarajan, R., Selvakumar, U. & Lingadurai, D. (2012), ‘Influence of search algorithms on aerodynamic design optimisation of aircraft wings’, *International Conference on Modelling Optimization and Computing* pp. 2155–2163.
- Omara, H. & Abidob, M. (2010), ‘Designing integrated guidance law for aerodynamic missiles by hybrid multi-objective evolutionary algorithm and tabu search’, *Aerospace Science and Technology* **14**(5), 356–363.
- Palacios, F., Colonno, M., Aranake, A., Campos, A., Copeland, S., Economon, T., Lonkar, A., Lukaczyk, T., Taylor, T. R. & Alonso, J. (2013), ‘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* .
- Sacks, J., Welch, W., Mitchell, T. & Wynn, H. (1989), ‘Design and analysis of computer experiments’, *Statistical Science* **4**(4), 409–435.
- Simpson, T., Peplinski, J., Koch, P. & Allen, J. (2001), ‘Metamodels for computer-based engineering design: survey and recommendations’, *Engineering with Computers* **17**(2), 129–150.

Zitzler, E. (1999), Evolutionary algorithm for multi-objective optimization: methods and application, Master's thesis, Institut für Technische Informatik und Kommunikationsnetze  
Computer Engineering and Networks Laboratory, Zurich, Switzerland.