JOURNAL OF AIRCRAFT
Vol. , No. ,

Postprint of: Amrit A., Leifsson L., Koziel S.: Fast Multi-Objective Aerodynamic Optimization Using Sequential Domain Patching and Multifidelity Models JOURNAL OF AIRCRAFT (2020), pp.1-11
DOI: 10.2514/1.C035500

Fast Multi-Objective Aerodynamic Optimization Using Sequential Domain Patching and Multifidelity Models

Anand Amrit* and Leifur Leifsson†
Iowa State University, Ames, Iowa 50011
and
Slawomir Koziel‡
Reykjavik University, 102 Reykjavik, Iceland

https://doi.org/10.2514/1.C035500

Exploration of design tradeoffs for aerodynamic surfaces requires solving of multi-objective optimization (MOO) problems. The major bottleneck here is the time-consuming evaluations of the computational fluid dynamics (CFD) model used to capture the nonlinear physics involved in designing aerodynamic surfaces. This, in conjunction with a large number of simulations necessary to yield a set of designs representing the best possible tradeoffs between conflicting objectives (referred to as a Pareto front), makes CFD-driven MOO very challenging. This paper presents a computationally efficient methodology aimed at expediting the MOO process for aerodynamic design problems. The extreme points of the Pareto front are obtained quickly using single-objective optimizations. Starting from these extreme points, identification of an initial set of Pareto-optimal designs is carried out using a sequential domain patching algorithm. Refinement of the Pareto front, originally obtained at the level of the low-fidelity CFD model, is carried out using local response surface approximations and adaptive corrections. The proposed algorithm is validated using a few multi-objective analytical problems and an aerodynamic problem involving MOO of two-dimensional transonic airfoil shapes where the figures of interest are the drag and pitching moment coefficients. A multifidelity model is constructed using CFD model and control points parameterizing the shape of the airfoil. The results demonstrate that an entire or a part of the Pareto front can be obtained at a low cost when considering up to eight design variables.

Nomenclature

A	=	cross-sectional area, m^2
a_∞	=	speed of sound, m/s
C_d	=	drag coefficient, $d/(q_\infty c)$
C_l	=	lift coefficient, $l/(q_\infty c)$
C_m	=	pitching moment coefficient, $M/(q_\infty c^2)$
c	=	chord length, m
$\boldsymbol{c}$	=	low-fidelity model output
d	=	drag force, N
$\boldsymbol{d}$	=	trust-region radius, m
$\boldsymbol{f}$	=	high-fidelity model output
H	=	objective function value
$\boldsymbol{l}$	=	lower bound of $\boldsymbol{x}$
j	=	single-objective optimization iteration
k	=	multi-objective optimization iteration
l	=	lift force, N
M	=	pitching moment
M_∞	=	Mach number, V_∞/a_∞
N	=	total Pareto optimal solutions
N_c	=	number of low-fidelity model evaluations
N_f	=	number of high-fidelity model evaluations
n	=	number of design variables
q_∞	=	dynamic pressure, $1/2\rho_\infty V_\infty^2$
$\boldsymbol{s}$	=	surrogate model output
$\boldsymbol{u}$	=	upper bound of $\boldsymbol{x}$
V_∞	=	flow speed, m/s
$\boldsymbol{x}$	=	design variables, m
δ	=	trust-region radius
ε_H	=	norm of H from last two iterations
ε_x	=	norm of $\boldsymbol{x}$ from last two iterations
ε_δ	=	norm of δ from last two iterations
ρ_∞	=	density, $\mathrm{kg/m}^3$

Received 9 March 2019; revision received 13 September 2019; accepted for publication 11 February 2020; published online XX epubMonth XXXX.

*Graduate Student, Department of Aerospace Engineering, 537 Bissel Road. Student Member AIAA.

†Assistant Professor, Department of Aerospace Engineering, 537 Bissel Road. Senior Member AIAA.

‡Professor, Engineering Optimization and Modeling Center, School of Science and Engineering, Menntavegur 1; also Faculty of Electronics, Telecommunications and Informatics, Gdansk University of Technology, Gdansk, Poland. Senior Member AIAA.

I. Introduction

DESIGN of modern engineering systems often involves the use of accurate physics-based computational models. The fidelity of the simulations, for example, in terms of discretization density of the structure at hand, increases when there is a need for higher accuracy and capturing nonlinear physics and nonlinear interactions between system disciplines, which is often required in the case of new or unconventional systems [1]. Use of accurate physics-based computer simulations for design purposes can pose significant challenges due to 1) their high computational cost (often ranging from a few hours to days or even weeks on high-performance computing clusters); 2) a large number of design variables, constraints, and objectives (which may be conflicting); and 3) a large number of model evaluations needed by state-of-the-art optimization techniques. In this paper, an efficient approach is proposed to determine the best possible tradeoffs between conflicting objectives for design exploration with accurate simulations of the flow past aerodynamic surfaces, such as the wings of an aircraft or the rotor blades of helicopters and wind turbines.

In this paper, an efficient multifidelity framework is proposed for multi-objective aerodynamic design exploration. Our algorithm determines the entire Pareto quickly using two levels of physics-based aerodynamics models. In particular, two single-objective optimal designs are determined using a multifidelity trust-region optimization algorithm with a low-fidelity model derived by a simplified modeling method [2] and adaptation using output space mapping (SM) [3]. The Pareto optimal solutions spanning between the two single-objective optimal points are obtained by relocating sequentially from one end of the front to another followed by a refinement process. For this task, the sequential domain patching (SDP)

technique is adopted here from the work by Koziel and Bekasiewicz [4] in the area of electrical engineering and extended to aerodynamic design exploration. In [4], surrogate-based optimization (SBO) is performed using a local data-fit model [5], which is constructed based on low-fidelity model evaluations that are sparsely sampled in the vicinity of the current point on the Pareto front, and subsequently corrected using a single high-fidelity model evaluation and an additive correction [3,6].

The distinct features of the novel aerodynamic design exploration approach presented in this paper include the following: 1) two Pareto-optimal points spanning the portion of the Pareto front to be explored are identified at low cost using multifidelity methods and a single-objective optimization (SOO) algorithm; 2) depending on the computational budget, either the entire Pareto or a part of it can be explored; 3) the proposed approach does not use metaheuristic algorithms (such as multi-objective genetic algorithms [MOGAs] [7] and multi-objective evolutionary algorithms [MOEAs] [8]); and 4) gradient information is not used to determine the Pareto front.

This paper is organized as follows. The next section gives a broad overview of multi-objective design optimization approaches, including works in the area of aerodynamic design. The third section gives the details of the proposed aerodynamic design exploration algorithm. In the following section, the proposed approach is characterized using analytical problems and the aerodynamic design exploration of transonic airfoil shapes in viscous flow. The paper ends with conclusion and remarks on future work.

II. Background

Design exploration of aerodynamic surfaces with multiple conflicting objectives can be performed by 1) parametric search guided by expert knowledge, 2) minimization of the aggregate objective function using SOO algorithms, or 3) simultaneous minimization of multiple objectives using multi-objective optimization (MOO) routines. The first usually fails to yield optimal designs; however, it is frequently used to bypass difficulties pertinent to rigorous optimization. In the SOO approach, the aggregated objective function is typically formed by a linear combination of all objectives with the weighting coefficients set to express the designer preferences (see, e.g., [9–13]). If the objective and constraint gradients can be computed using adjoint sensitivity information [14], the SOO problem can be solved efficiently using the epsilon-constrained method [15]. The disadvantage is that only one Pareto-optimal design is found per SOO algorithm run, and the location of this design with respect to the overall Pareto front is unknown. Furthermore, the method is unable to identify nonconvex portions of the front.

In the MOO approach, the goal is to obtain designs representing the best possible tradeoffs between conflicting objectives. It is typically found in the form of a Pareto set (a discrete representation of the Pareto front) [11]. Population-based metaheuristics are the most popular solution approaches [7,8,16–20]. Examples include evolutionary algorithms (EAs) [16], MOGAs [7], and MOEAs [8]. Other multi-objective metaheuristic approaches include particle swarm optimization [17], differential evolution [18], firefly [19], and cuckoo search [20]. Unlike the SOO approach, these population-based techniques are capable of generating the entire Pareto set in a single algorithm run. Unfortunately, metaheuristics require numerous model evaluations, which limits their use to aerodynamic design exploration problems of low complexity (i.e., problems with a low number of designable parameters and simulations with a low number of degrees of freedom).

Surrogate-based optimization (SBO) [5,21] techniques have recently become popular as a means of addressing the high computational cost of the optimization process. The main steps of the SBO procedure are 1) sampling the design space using a design of experiments technique of choice (e.g., [22,23]), 2) acquiring the training data through high-fidelity model simulations, 3) constructing the surrogate model using the observations, and 4) updating the surrogate through allocation of additional samples. Popular surrogate modeling approaches include response surface approximations [5], radial-based function models [21], and kriging interpolation [21]. New samples (so-called infill points) are assigned using appropriate infill criteria [21], which may involve identification of approximated optimal designs (such as Pareto-optimal solutions in case of MOO).

The SBO process plays a central role in surrogate-assisted MOO algorithms. Examples of such algorithms include Pareto-based efficient global optimization [24] (ParEGO), which uses the weighted-sum approach, and the Pareto set pursuing (PSP) approach [25], which uses global surrogate models for MOO. A global surrogate-assisted MOO with constraints based on expected improvement of the objective functions is described in [26]. The inexact pre-evaluation approach [27] is extended in [28] for MOEAs. Locally constructed radial-basis function models, adaptive sampling, and surrogate modeling are combined for MOGAs in [29]. A global approximation-based MOO for robust design under interval uncertainty is described in [30]. Finally, surrogate-assisted MOO using global and local models is introduced in [31].

Recent applications of surrogate-assisted MOO algorithms for aerodynamic shape optimization involve various combinations of metaheuristics and surrogate modeling methods. Zhang et al. [32] used computational fluid dynamics (CFD) models and kriging surrogate models along with the NSGA-II algorithm to perform MOO on high-speed train head shapes. Wang et al. [33] performed an aerodynamic MOO to maximize the pressure ratio and adiabatic efficiency of compressor rotors using response surface models and MOGA [7]. Leusink et al. [34] performed aerodynamic MOO design of helicopter blades with MOGA using Gaussian process regression techniques [21] to construct the surrogate models. Amrit et al. [8] and Leifsson et al. [35,36] used MOEAs, kriging surrogate models, and design space confinement strategies to perform MOO of transonic airfoil shapes. Fincham and Friswell [37] used MOGA and radial-basis functions models [5,21] to represent aerodynamic surfaces and performed a multi-objective aerodynamic shape optimization of camber morphing airfoil shapes.

Multifidelity methods [21,38] use information from models of varying degree of fidelity to leverage the computational speed-up of the low-fidelity models and the accuracy of the high-fidelity ones. A typical approach is to use the fast low-fidelity models to accelerate the design optimization process and yield initial approximations of the optimum designs, which is followed by (usually) iterative references to the high-fidelity models (through various model management strategies) aimed at refinement of the solution accuracy.

Low-fidelity modeling approaches include simplified modeling methods (e.g., simplified governing equations [39] and coarse discretization [2]), projection-based methods (e.g., proper orthogonal decomposition [40] and reduced basis method [41]), and data-fit methods (e.g., radial basis functions [5], kriging [21], and support vector regression [42]).

Model management strategies include adaptation, fusion, and filtering. Adaptation approaches can be divided into global methods (e.g., efficient global optimization [EGO] using global data-fit models and infill criteria based on expected improvement to balance exploitation and exploration [21]) or local methods (e.g., SBO methods using local data-fit models [21] and multifidelity trust-region methods using corrected low-fidelity models with the corrections classified as additive [6], multiplicative [43], comprehensive [44], or SM [3]). Fusion approaches evaluate the low- and high-fidelity models on a given set of samples and subsequently combine the outputs in one model (e.g., cokriging [45] and Bayesian regression [46]). In filtering methods, the high-fidelity model is invoked following the evaluation of a low-fidelity filter (e.g., the multifidelity stochastic collocation approach [47]). Multifidelity methods have been applied successfully to single-objective aerodynamic design problems (see, e.g., [2,39,43]). The use of multifidelity models for multi-objective aerodynamic design problems, however, is not as well studied. This work presents the application and extension of the SDP technique [4] to aerodynamic design exploration.

III. Methodology

This section describes design exploration using SDP for Pareto set identification and gives the details of the MOO framework as well as multifidelity modeling.

A. Definition of the Pareto Front

Here, the concept of Pareto front is explained using a specific example of an aerodynamic design problem. The goal is to find a tradeoff between various aerodynamic forces, such as lift, drag, and pitching moment coefficients, denoted as $C_{l.f}$, $C_{d.f}$, and $C_{m.f}$, respectively. Let an accurate high-fidelity aerodynamics simulation model be denoted as $\boldsymbol{f}(\boldsymbol{x}) = [\, C_{l.f}(\boldsymbol{x}) \quad C_{d.f}(\boldsymbol{x}) \quad C_{m.f}(\boldsymbol{x}) \,]^T$, where $\boldsymbol{x}$ is the $n \times 1$ vector of design variables.

Let $F_k(\boldsymbol{x}), k = 1, \ldots, N_{\text{obj}}$, be a kth design objective of interest. If $N_{\text{obj}} > 1$ then any two designs, $\boldsymbol{x}^{(1)}$ and $\boldsymbol{x}^{(2)}$ for which $F_k(\boldsymbol{x}^{(1)}) < F_k(\boldsymbol{x}^{(2)})$ and $F_l(\boldsymbol{x}^{(2)}) < F_l(\boldsymbol{x}^{(1)})$ for at least one pair $k \neq l$, are not commensurable; that is, none is better than the other in the multi-objective sense. We define a Pareto dominance relation $\prec$, saying that for the two designs $\boldsymbol{x}$ and $\boldsymbol{y}$, we have $\boldsymbol{x} \prec \boldsymbol{y}$ ($\boldsymbol{x}$ dominates $\boldsymbol{y}$) if $F_k(\boldsymbol{x}) \leq F_k(\boldsymbol{y})$ for all $k = 1, \ldots, N_{\text{obj}}$, and $F_k(\boldsymbol{x}) < F_k(\boldsymbol{y})$ for at least one k [48]. The goal of the MOO is to find a representation of a so-called Pareto front (of Pareto-optimal set) $\boldsymbol{X}_P$ of the design space $\boldsymbol{X}$, such that for any $\boldsymbol{x} \in \boldsymbol{X}_P$, there is no $\boldsymbol{y} \in \boldsymbol{X}$ for which $\boldsymbol{y} \prec \boldsymbol{x}$.

B. Pareto Front Exploration Using Sequential Domain Patching

The proposed Pareto front exploration approach is based on the SDP algorithm proposed by Koziel and Bekasiewicz [4], and it is adopted and applied in this work for the multifidelity aerodynamic design. The approach proposed in this paper is formulated in terms of two scalar design objectives, F_1 and F_2, and produces a sequence of designs, $\boldsymbol{x}^{(k)*}$, $k = 1, 2, \ldots, N$, where $\boldsymbol{x}^{(1)*}$ and $\boldsymbol{x}^{(N)*}$ are the two ends of the Pareto front to be explored and N is the total number of Pareto optimal points. To obtain the entire Pareto set, initially, two points representing the extreme Pareto-optimal solutions are obtained by minimizing individual objectives, one at a time, as shown in Fig. 1a. If, instead of the entire front, only its part needs to be explored, then two SOOs are carried out for one of the objective functions while subjecting the second objective function into a nonlinear constraint to obtain two target points on the front, as indicated in Fig. 1b. An alternative method is a weighted-sum approach [13] that can be used to obtain two points on the Pareto front per the designer preferences encoded in the weighting factors.

Once the extreme ends of the Pareto front to be explored are obtained, the SDP-based MOO algorithm is executed, as explained in Sec. II.C, to obtain an initial Pareto set (see Fig. 2). The optimal solutions in the initial Pareto are explored within the patches constructed in the vicinity of the starting points as shown in Fig. 2a. The procedure continues until the entire distance between $\boldsymbol{x}^{(1)*}$ and $\boldsymbol{x}^{(N)*}$ has been traversed. Because of the high cost of the high-fidelity model involved in the multi-objective aerodynamic problem, the algorithm is designed so as to obtain the initial Pareto set at the level of an auxiliary low-fidelity model. Subsequently, refinement of the initial Pareto set is performed using a limited number of high-fidelity model evaluations and response surface approximation models to leverage the accuracy of the final Pareto set (Fig. 2a). This process is explained in detail in Secs. II.C–II.E.

C. Multifidelity, Multi-Objective Aerodynamic Sequential Domain Patching Algorithm

This section describes the proposed SDP-based aerodynamic MOO algorithm in detail. The first extreme end of the Pareto front is a solution to the SOO problem of the form

$$\boldsymbol{x}^{(1)*} = \arg \min_{l \leq x \leq u} F_1(\boldsymbol{x}) \tag{1}$$

subject to

$$g(\boldsymbol{x}) \leq 0$$

where $g(\boldsymbol{x})$ stands for the inequality constraints for the problem at hand. $\boldsymbol{x}^{(N)*}$ is obtained in a similar manner by minimizing F_2 subjected to appropriate inequality constraint.

The cost of solving Eq. (1) can be high depending on the dimension of the problem and the cost of the model evaluations. To expedite the process of solving Eq. (1), a trust-region-based multifidelity optimization algorithm [39] is executed. The multifidelity model is constructed using output SM [49] in this work. A combination of the accurate high-fidelity model $\boldsymbol{f}$ and a model $\boldsymbol{c}$, which is of lower fidelity than $\boldsymbol{f}$ and computationally faster to evaluate, is exploited by

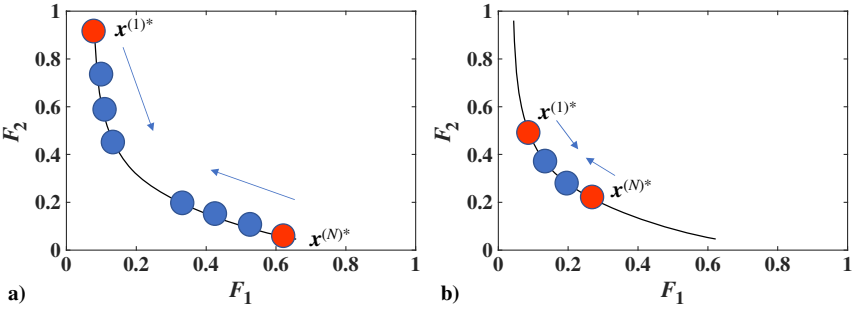

Fig. 1 Exploration of a) the entire Pareto front and b) only part of the Pareto front.

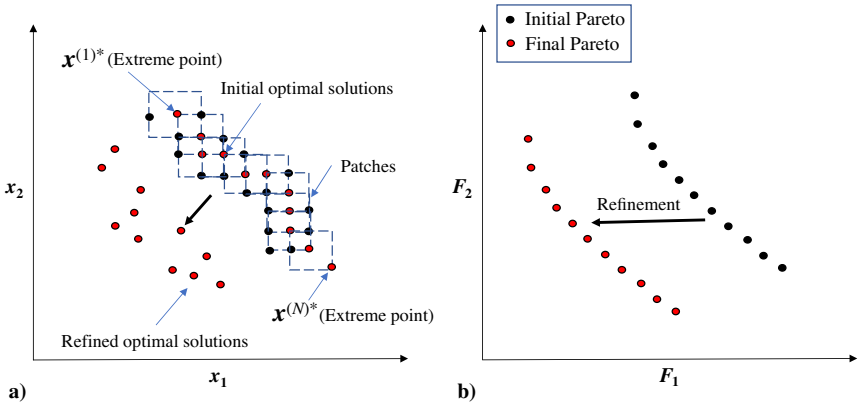

Fig. 2 Pictorial representation of the sequential domain patching method: a) patches in the design space; b) refinement of the Pareto front in the objective space.

the output SM. Here, the low-fidelity model $\boldsymbol{c}$ is based on coarse-discretization CFD simulations (see, e.g., [39] for a discussion on approaches for low-fidelity modeling). The output SM algorithm produces a sequence $\boldsymbol{x}^{(1,j)}$, $j = 0, 1, \ldots$, of approximate solutions to Eq. (1) as [49]

$$\boldsymbol{x}^{(1,j+1)} = \arg \min_{\boldsymbol{x}, \|\boldsymbol{x}^{(1,j)} - \boldsymbol{x}^{(1,j-1)}\| \le \delta^{(j)}} F_1(\boldsymbol{s}^{(1,j)}(\boldsymbol{x})) \tag{2}$$

where $\boldsymbol{s}^{(1,j)}(\boldsymbol{x}) = \left[C_{l.s}^{(1,j)}(\boldsymbol{x}) \quad C_{d.s}^{(1,j)}(\boldsymbol{x}) \quad C_{m.s}^{(1,j)}(\boldsymbol{x}) \right]^T$ is the surrogate model at iteration j. The output SM surrogate model is

$$\boldsymbol{s}^{(1,j)}(\boldsymbol{x}) = \boldsymbol{A}^{(1,j)} \circ \boldsymbol{c}(\boldsymbol{x}) + \boldsymbol{D}^{(1,j)} \tag{3}$$

where $\circ$ denotes componentwise multiplication, and the multiplicative and additive terms, $\boldsymbol{A}^{(1,j)}$ and $\boldsymbol{D}^{(1,j)}$, respectively, are calculated analytically. For the drag coefficient C_d the terms are calculated as

$$\begin{bmatrix} a_d^{(1,j)} \\ d_d^{(1,j)} \end{bmatrix} = (\boldsymbol{C}_d^T \boldsymbol{C}_d)^{-1} \boldsymbol{C}_d^T \boldsymbol{F}_d \tag{4}$$

$$\boldsymbol{C}_d = \begin{bmatrix} C_{d.c}(\boldsymbol{x}^{(1,0)}) & C_{d.c}(\boldsymbol{x}^{(1,1)}) \ldots C_{d.c}(\boldsymbol{x}^{(1,j)}) \\ 1 & 1 \ldots 1 \end{bmatrix}^T \tag{5}$$

$$\boldsymbol{F}_d = \left[C_{d.f}(\boldsymbol{x}^{(1,0)}) C_{d.f}(\boldsymbol{x}^{(1,1)}) \ldots C_{d.f}(\boldsymbol{x}^{(1,j)}) \right]^T \tag{6}$$

where $C_{d.c}$ and $C_{d.f}$ represent the drag coefficient values obtained by evaluations of the low- and high-fidelity models, respectively. Similar models are constructed for C_m and C_l.

Using the SOO points, the MOO algorithm for the initial Pareto front representation is executed and can be formally summarized as follows [4]:

1) Patch size $\boldsymbol{d} = [d_1, \ldots, d_n]^T$ is set using the procedure of Sec. II.D.

2) Current points are set as $\boldsymbol{x}_{c1} = \boldsymbol{x}^{(1)*}$ and $\boldsymbol{x}_{cN} = \boldsymbol{x}^{(N)*}$.

3) n perturbations of the size $\boldsymbol{d}$ are evaluated around $\boldsymbol{x}_{c1}$ (toward $\boldsymbol{x}_{cN}$ only), and the one that brings the largest improvement with respect to the second objective F_2 is selected.

4) The patch is relocated so that it is centered at the best perturbation selected in step 3; $\boldsymbol{x}_{c1}$ is updated.

5) n perturbations of the size $\boldsymbol{d}$ are evaluated around $\boldsymbol{x}_{cN}$ (toward $\boldsymbol{x}_{c1}$ only), and the one that brings the largest improvement with respect to the second objective F_1 is selected.

6) The patch is relocated so that it is centered at the best perturbation selected in step 5; $\boldsymbol{x}_{cN}$ is updated.

7) If the path between $\boldsymbol{x}^{(1)*}$ and $\boldsymbol{x}^{(N)*}$ is not complete, go to step 3.

The flowchart shown in Fig. 3 outlines the Pareto front exploration procedure using the SDP algorithm and multifidelity aerodynamic models. The major differences of the proposed algorithm and the one presented in [4] lie in a) the use of the multifidelity modeling (3–6), due the highly nonlinear aerodynamic models, used for solving the problem (2) and obtaining the SOO points, and b) the exploration of designs outside the design space enclosed within the patches of the SOO points (cf. Fig. 2).

The algorithm yields a set of patches, covering a part of design space that contains the initial approximation of the set of Pareto-optimal solutions. The total computational cost of the algorithm depends on n and on the total number of patches. The net cost can be computed as $(M-1)\cdot(n-1)$, which excludes the cost of solving Eq. (1), where $M = \sum_{k=1,\ldots,n} m_k$ and is the number of intervals in the direction j. However, in practice, the cost can be lower as some perturbations may not be evaluated due to the imposed constraints. Here, we describe in detail the step-by-step procedure of obtaining the Pareto front:

Step 1: Before the algorithm is initialized, two SOOs are performed from some random design within the design bounds. The SOO problems are solved using the trust-region, multifidelity algorithm, and the SM model [49]. The solutions from the SOO problems

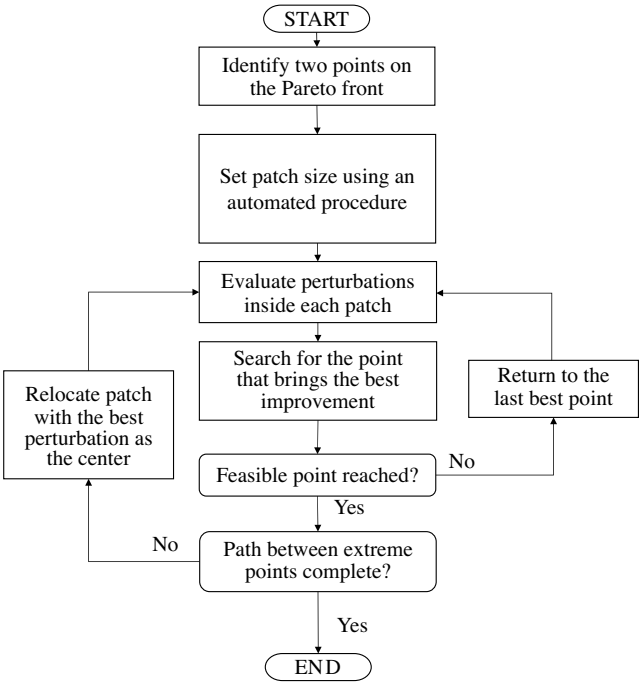

Fig. 3 Aerodynamic design exploration based on sequential domain patching.

are used as input to the automated domain patching algorithm explained in Sec. II.D to obtain the patch size $\boldsymbol{d}$.

Step 2: Solutions of the SOO problems are used as the starting points for the algorithm as marked in Fig. 2a as $\boldsymbol{x}^{(1)*}$ and $\boldsymbol{x}^{(N)*}$. The next points are searched for in the vicinity of these starting points while moving in either direction, that is, from $\boldsymbol{x}^{(1)*}$ to $\boldsymbol{x}^{(N)*}$ or vice versa.

Step 3: A patch is constructed with n perturbations of size $\boldsymbol{d}$ around $\boldsymbol{x}^{(1)*}$. Each perturbation is evaluated on the low-fidelity model to obtain objective functions and constraints values. The design brings the largest improvement with respect to the objective F_1. The search for largest improvement in F_1 is performed with a given condition that the designs are well within the global bounds and also they satisfy the constraints. The algorithm is designed specifically for a design space that has all feasible designs. However, to satisfy the constraints, a surrogate model similar to Eq. (2) can be used.

Step 4: The best perturbation result obtained from step 3 is used to update step 2; that is, $\boldsymbol{x}_{c1}$ is updated. The patch is relocated so that the center of the patch is the updated $\boldsymbol{x}_{c1}$.

Step 5: A patch is constructed with n perturbations of size $\boldsymbol{d}$ around $\boldsymbol{x}^{(N)*}$. Each perturbation is evaluated on the low-fidelity model to obtain objective function and constraint values. The design brings the largest improvement with respect to the objective F_1. The search for largest improvement in F_1 is performed with a given condition that the designs are well within the global bounds and also they satisfy the linear and nonlinear constraints.

Step 6: The best perturbation result obtained from step 5 is used to update step 2; that is, $\boldsymbol{x}_{cN}$ is updated. The patch is relocated so that the center of the patch is the updated $\boldsymbol{x}_{cN}$.

Step 7: Steps 2–6 are continued until the path between $\boldsymbol{x}^{(1)*}$ and $\boldsymbol{x}^{(N)*}$ is complete.

D. Automated Determination of Patch Sizes

Using an automated technique to determine patch size, similar to the one presented in [4], the distance between $\boldsymbol{x}^{(1)*}$ and $\boldsymbol{x}^{(N)*}$ is split into an integer-valued number of intervals. The number of intervals in each direction of the design variable given by m_k is assigned by the following procedure, where we use the notation $\boldsymbol{x}^{(1)*} = \left[\boldsymbol{x}_1^{(1)*}, \ldots, \boldsymbol{x}_n^{(1)*} \right]^T$ (similarly for $\boldsymbol{x}^{(N)*}$):

1) $F(\boldsymbol{c})$ is evaluated at n points $\boldsymbol{x}_k^{(1-N)*} = \left[\boldsymbol{x}_1^{(1)*} \ldots \boldsymbol{x}_{k-1}^{(1)*} \; \boldsymbol{x}_k^{(N)*} \; \boldsymbol{x}_{k+1}^{(1)*} \ldots \boldsymbol{x}_n^{(1)*} \right]^T$, $k = 1, \ldots, n$; where n is the design space dimensionality.

2) Calculate $E_{1.k} = \left\| \boldsymbol{c}\left(\boldsymbol{x}_k^{(1-N)*}\right) - \boldsymbol{c}\left(\boldsymbol{x}^{(1)*}\right) \right\| \Big/ \left\| \boldsymbol{c}\left(\boldsymbol{x}^{(1)*}\right) \right\|$, $k = 1, \ldots, n$.

3) $F(\boldsymbol{c})$ is evaluated at n points $\boldsymbol{x}_k^{(N-1)*} = \left[\boldsymbol{x}_1^{(N)*} \ldots \boldsymbol{x}_{k-1}^{(N)*}\ \boldsymbol{x}_k^{(1)*}\ \boldsymbol{x}_{k+1}^{(N)*} \ldots \boldsymbol{x}_n^{(N)*}\right]^T$, $k = 1, \ldots, n$.

4) Calculate $E_{N.k} = \left\| \boldsymbol{c}\left(\boldsymbol{x}_k^{(N-1)*}\right) - \boldsymbol{c}\left(\boldsymbol{x}^{(N)*}\right) \right\| \Big/ \left\| \boldsymbol{c}\left(\boldsymbol{x}^{(N)*}\right) \right\|$, $k = 1, \ldots, n$.

5) Set $E_k = (E_{1.k} + E_{N.k})/2$.

6) Normalize $E_k = E_k / \max\{E_j : j = 1, \ldots, n\}$.

7) Set $m_k = \max\{2, m_{\max}.E_k\}$, $k = 1, \ldots, n$.

Varying the kth component of $\boldsymbol{x}^{(1)*}$ toward $\boldsymbol{x}^{(N)*}$ gives relative response changes, $E_{1.k}$ (similarly for $E_{N.k}$). The value of m_k is rounded to a nearest integer with the minimum value 2 as the default. The maximum number of intervals per geometric direction, $m_{\max}$, is a user-defined parameter and can be set based on a maximum allowed relative response change $E_{\max}$ as follows: $m_{\max} = \lceil \max\{E_k : k = 1, \ldots, n\}/E_{\max} \rceil$ (calculated for unnormalized E_k factors). In case we have a specific computational budget, the value of $m_{\max}$ can be adjusted as per requirement.

E. Pareto Set Refinement

The algorithm discussed in Sec. II.C is used to determine the initial Pareto at the level of the low-fidelity model $\boldsymbol{c}$. To obtain the high-fidelity Pareto-optimal designs $\boldsymbol{x}_f^{(k)}$, $k = 1, \ldots, N$, the following procedure is executed:

$$\boldsymbol{x}_f^{(k)} \leftarrow \arg \min_{\boldsymbol{x}, F_2(\boldsymbol{x}) \le F_2(\boldsymbol{x}_f^{(k)})} F_1\left(\boldsymbol{s}_q(\boldsymbol{x}) + \left[\boldsymbol{f}\left(\boldsymbol{x}_f^{(k)}\right) - \boldsymbol{s}_q\left(\boldsymbol{x}_f^{(k)}\right)\right]\right) \tag{7}$$

In this refinement process, the first objective is improved without degrading the second objective. The above process begins with $\boldsymbol{x}_f^{(k)} = \boldsymbol{x}_c^{(k)}$ as the starting point and the process is iterated until convergence. The correction term in Eq. (7) makes sure that $\boldsymbol{s}_q(\boldsymbol{x}_f^{(k)}) = \boldsymbol{f}(\boldsymbol{x}_f^{(k)})$ at the initiation of each iteration. The surrogate model $\boldsymbol{s}_q$ used in this process is a second-order polynomial approximation without the mixed terms. The approximation model is based on the low-fidelity model $\boldsymbol{c}$ evaluated at $\boldsymbol{x}_c^{(k)}$ and the perturbed designs within the patch surrounding $\boldsymbol{x}_c^{(k)}$.

IV. Numerical Examples

In this section, the proposed algorithm is demonstrated using two analytical problems and a two-dimensional multi-objective aerodynamic design problem.

A. Analytical Problems

The analytical problems, the Fonseca and Fleming function [50] and the Zitzler–Deb–Thiele's function $N.1$ (ZDT 1) [51], are used to demonstrate the application of the proposed algorithm.

1. *Zitzler–Deb–Thiele's Function N.1 (ZDT 1)*

The formulation of the test problem is given by

$$\min\ f_1 = x_1 \tag{8}$$

$$\min\ f_2 = u\left(1 - \sqrt{x_1/u}\right) \tag{9}$$

where

$$x_i \in [0, 1], \qquad i = 1, \ldots, 30$$

and $u = 1 + 9/7 \sum_{i=2}^{30} x_i$.

The analytical functions f_1 and f_2 are considered as the high-fidelity model $\boldsymbol{f}$. A low-fidelity model $\boldsymbol{c}$ is formulated by adding noise (Δf) to the analytical functions as

$$f_{1.c} = f_1 + \Delta f \tag{10}$$

$$f_{2.c} = f_2 + \Delta f \tag{11}$$

where $\Delta f = 0.1 x_1 + 0.5$.

Figure 4 shows the characteristic features of low- and high-fidelity models. The approach explained in Sec. II is executed to obtain the extreme points of the Pareto using SOO and then yield the entire Pareto front.

Figure 5 shows the initial and final Pareto fronts. The initial Pareto front (shown in black dots) is obtained using the low-fidelity model and the algorithm in Sec. II.C. The final Pareto front (shown in red dots) is then obtained after executing the refinement procedure given in Sec. II.E. The proposed algorithm used 50 low-fidelity model evaluations and 24 high-fidelity model evaluations. The exact analytical solution (shown by the blue dashed curve) compares well with the results of the proposed algorithm for this case.

2. *Fonseca and Fleming Function*

The formulation of the test problem is given by

$$\min\ f_1 = 1 - \exp\left[-\sum_{i=1}^{n}\left(x_i - \frac{1}{\sqrt{n}}\right)^2\right] \tag{12}$$

$$\min\ f_2 = 1 - \exp\left[-\sum_{i=1}^{n}\left(x_i + \frac{1}{\sqrt{n}}\right)^2\right] \tag{13}$$

where

$$x_i \in [-4, 4], \qquad i = 1, \ldots, 8$$

The analytical functions f_1 and f_2 are considered as the high-fidelity accurate model $\boldsymbol{f}$. The low-fidelity model $\boldsymbol{c}$ is formulated by adding noise (Δf) to the analytical functions in the same way as in Eq. (11).

The initial and final Pareto fronts obtained using the proposed algorithm are given in Fig. 6. The initial Pareto front is shown in black dots, and the final Pareto front is shown in red dots. The proposed algorithm used 150 low-fidelity model evaluations and 16 high-fidelity model evaluations. The results compare well with the exact analytical solution for this case, which is shown by the blue dashed curve.

B. Transonic Airfoil Design

This section demonstrates the proposed algorithm for the multi-objective design optimization of an airfoil in transonic flow.

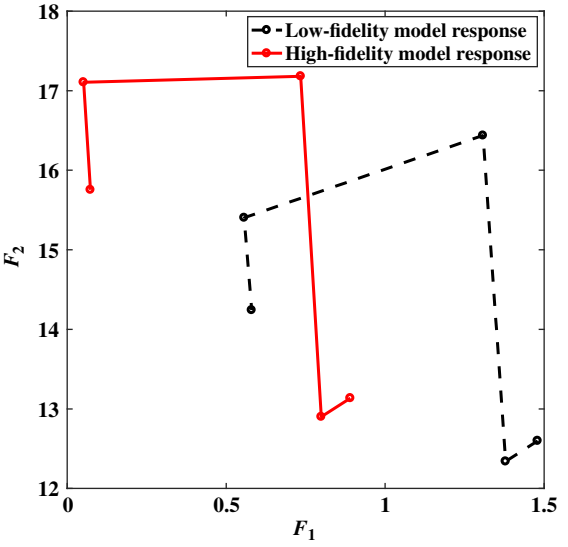

Fig. 4 Representation of high- and low-fidelity evaluation values of five random designs.

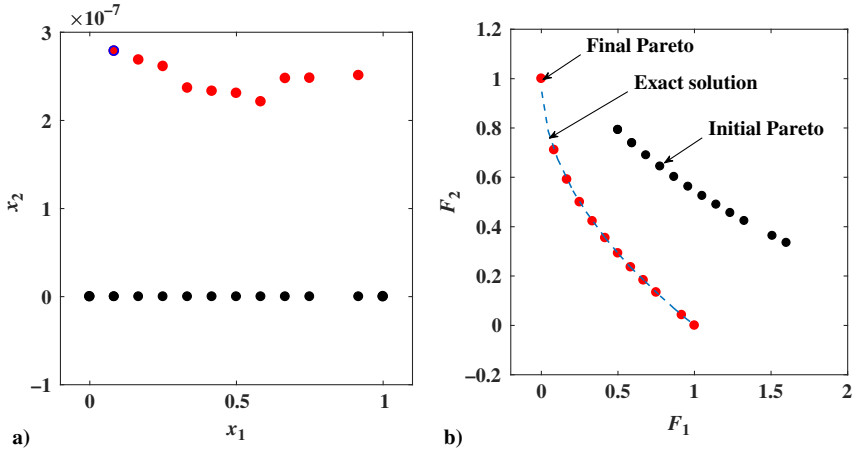

Fig. 5 Pareto front obtained for the ZDT 1 function: a) design space; b) feature space.

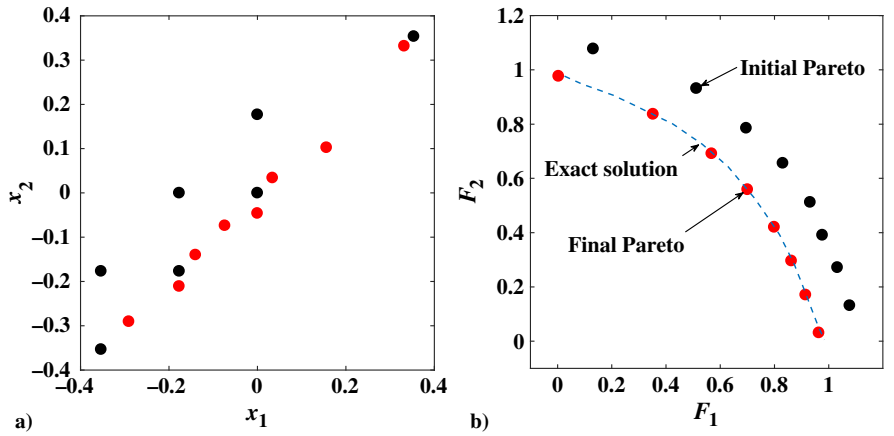

Fig. 6 Pareto front obtained for the Fonseca and Fleming function: a) design space; b) feature space.

1. *Problem Description*

The main goal of the aerodynamic problem is to obtain the trade-offs between conflicting objectives, the drag coefficient C_d and the pitching moment coefficient C_m, of the RAE 2822 at a freestream Mach number of $M_\infty = 0.734$, lift coefficient C_l of 0.824, and Reynolds number (Re) of 6.5×10^6, subject to a cross-sectional area (A) constraint. Specifically, the conflicting objectives considered here are the high-fidelity values of the drag coefficient and the pitching moment coefficient; that is, we have $F_1(\boldsymbol{x}) = C_{d.f}$ and $F_2(\boldsymbol{x}) = C_{m.f}$, and the multi-objective constrained optimization problem can be expressed as:

$$\min_{l \le x \le u} C_d, \qquad \max_{l \le x \le u} C_m \tag{14}$$

subject to

$$C_l(\boldsymbol{x}) = 0.824$$

and $A(\boldsymbol{x}) \geq A_{\text{baseline}}$, where A_{baseline} is the cross-sectional area of the baseline RAE2822 airfoil. It should be noted here that this formulation is a modified version of the single-objective benchmark aerodynamic design optimization case developed by the AIAA Aerodynamic Design Optimization Discussion Group (see, e.g., [4], for the original formulation).

2. *Design Variables*

The airfoil shape is controlled using the B-spline parameterization approach described by Ren et al. [49]. Figure 7a shows eight control points, four on each of the top and bottom surfaces, that can move in the vertical direction. The leading and trailing edge endpoints of the airfoil are fixed in all directions. The x locations of the eight control points (eight design variables) are based on a fit to the RAE 2822 as

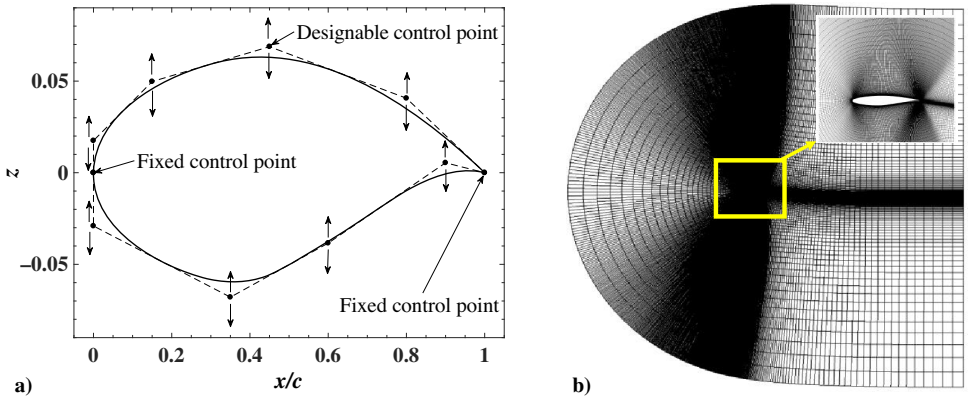

Fig. 7 Airfoil computational models: a) airfoil shape parameterization; b) hyperbolic C-mesh.

$\boldsymbol{X} = [\boldsymbol{X}_u;\ \boldsymbol{X}_l]^T = [0.0\ \ 0.15\ \ 0.45\ \ 0.80;\ \ 0.0\ \ 0.35\ \ 0.60\ \ 0.90]^T$, and the initial design variable vector is $\boldsymbol{x} = [\boldsymbol{x}_u;\ \boldsymbol{x}_l]^T = [0.0175\ \ 0.0498\ \ 0.0688\ \ 0.0406;\ \ -0.0291\ \ -0.0679\ \ -0.0384\ \ 0.0054]^T$. The lower bound of $\boldsymbol{x}$ is set as $\boldsymbol{l} = [0.015\ \ 0.015\ \ 0.015\ \ 0.015;\ \ -0.08\ \ -0.08\ \ -0.08\ \ -0.01]^T$, and the upper bound is set as $\boldsymbol{u} = [0.08\ \ 0.08\ \ 0.08\ \ 0.08;\ \ -0.01\ \ -0.015\ \ -0.015\ \ 0.01]^T$.

3. *High-Fidelity Viscous Aerodynamics Model*

The physics problem involves solving a viscous case using Stanford University Unstructured (SU^2) [52] implicit density-based flow solver. The high-fidelity aerodynamic model ($\boldsymbol{f}$) solves the steady compressible Reynolds-averaged Navier–Stokes (RANS) equations with the Spalart–Allmaras turbulent model [53] using the SU^2 implicit density-based solver. The second-order Jameson-Schmidt-Turkel scheme [54] is used to calculate the convective flux along with one multigrid level to accelerate the solution. The turbulent variables are convected using a first-order scalar upwind method. The flow solver convergence criterion is the one that occurs first of the two: 1) the change in the drag coefficient value over the last 100 iterations is less than 10^{-5}, or 2) a maximum number of iterations of 20,000 is met.

The computational grid is generated using the hyperbolic C-mesh of Kinsey and Barth [55] (see Fig. 7b) with the far-field set 100 chord lengths from the airfoil surface. To have the wall y^+ values within reasonable values (i.e., $y^+ < 5$) around the airfoil surface, the distance from the airfoil surface to the first node is $10^{-5}c$, where c is the airfoil chord length. The grid points are clustered at the trailing edge and the leading edge of the airfoil with the density controlled by the number of points in the streamwise direction and in the direction normal to the airfoil surface.

Table 1 gives the results of a grid convergence study using the RAE 2822 airfoil at $M_\infty = 0.734$, $C_l = 0.824$, and $Re_\infty = 6.5 \times 10^6$. The constant lift condition is achieved by internally changing the angle of attack within the flow solver. It can be seen that the difference between meshes 3 and 4 is 1.8 drag counts (note that we define one drag count [d.c.] as $\Delta C_d = 10^{-4}$, and one lift count [l.c.] as $\Delta C_l = 10^{-2}$), which is not within the standard expectation of a converged solution mesh density, that is, less than 0.1 d.c. There is, however, a significant difference in the simulation times, that is, 153 minutes for mesh 4 and 34 minutes for mesh 3. Therefore, for the purpose of demonstrating the proposed algorithm, mesh 3 is chosen for the high-fidelity model $\boldsymbol{f}$ simulations.

4. *Low-Fidelity Viscous Aerodynamics Model*

The model setup for the low-fidelity model is same as that of the high-fidelity model $\boldsymbol{f}$, with the grid density being far less than that of the high-fidelity one. As shown in Table 1, we use Mesh 1 for the low-fidelity model $\boldsymbol{c}$. The low-fidelity model convergence criteria are set with the following values occurring first: 1) the change in the drag coefficient value over the last 100 iterations is less than 10^{-4}, or 2) the maximum number of iterations is set to 5000.

5. *Single-Objective Optimization Results*

Two SOO problems are solved using the SM algorithm [3,49] as described in Sec. II.B, and the results are shown in Table 2. Algorithm 1 is executed iteratively using trust-region, gradient-based optimization to obtain the initial point (cf. Fig. 1). The gradient-based search uses the sequential quadratic programming algorithm, where the original problem is solved iteratively by replacing the original objective function

Table 1 Grid convergence study for the baseline shape

Mesh	Number of elements	Lift counts	Drag counts	Simulation time,[a] min
1	9,836	82.4	324.6	3.1
2	38,876	82.4	221.5	8.8
3	154,556	82.4	204.8	34.0
4	616,316	82.4	203.0	152.6

[a]Computed on a high-performance cluster with 32 processors. Flow solution only.

Table 2 Single-objective optimization results

Parameter/method	Baseline	SOO problem 1	SOO problem 2
C_l (l.c.)	82.35	82.39	82.39
C_d (d.c.)	203.80	116.9	121.8
$C_{m,c/4}$	−0.0905	−0.1023	−0.0736
A	0.0779	0.0779	0.0779
N_c	— —	550	499
N_f	— —	4	4
CPU time, h	— —	14	13.55

(and nonlinear constraints) by their respective local quadratic models (linear for constraint functions). To obtain the gradient information, finite differences are used on the surrogate model $\boldsymbol{s}(\boldsymbol{x})$ with the finite difference step size set at 10^{-5}. Once the local optimum is reached, the design is evaluated on the high-fidelity model $\boldsymbol{f}$ and fed into the gradient-based algorithm to search for further minima, and the process is continued until convergence. Optimization of the multifidelity model is constrained to the vicinity of the current design defined as $\|\boldsymbol{x} - \boldsymbol{x}^{(i)}\| \le \delta^{(i)}$, with the trust-region radius $\delta^{(i)}$ adjusted adaptively using the standard trust-region rules [56]. The convergence tolerances for the termination conditions are set as $\varepsilon_x = 10^{-4}$, $\varepsilon_H = 10^{-4}$, and $\varepsilon_\delta = 10^{-4}$.

Because of the computational expense of the CFD models, instead of exploring the entire Pareto, that is, in between the optimal points of the two objective functions (the extreme points), only a part of the Pareto front is explored. The algorithm is run to obtain the Pareto in between $C_m = -0.074$ and $C_m = -0.11$ (chosen here for illustration purposes). Subsequently, two SOO problems are solved to obtain the endpoints of the Pareto front:

SOO problem 1:

$$\boldsymbol{x}^{(1)*} = \arg \min_{\boldsymbol{x}, \|\boldsymbol{x}-\boldsymbol{x}^{(i)}\| \le \delta^{(i)}} C_d(\boldsymbol{s}(\boldsymbol{x})) \tag{15}$$

subjected to $C_l(\boldsymbol{x}) = 0.824$,

$$C_m(\boldsymbol{x}) \ge -0.11 \quad A(\boldsymbol{x}) \ge A_{\text{baseline}}$$

SOO problem 2:

$$\boldsymbol{x}^{(N)*} = \arg \min_{\boldsymbol{x}, \|\boldsymbol{x}-\boldsymbol{x}^{(i)}\| \le \delta^{(i)}} C_d(\boldsymbol{s}(\boldsymbol{x})) \tag{16}$$

subjected to $C_l(\boldsymbol{x}) = 0.824$,

$$C_m(\boldsymbol{x}) \ge -0.074 \quad A(\boldsymbol{x}) \ge A_{\text{baseline}}$$

where $\boldsymbol{s}(\boldsymbol{x})$ is a fast surrogate model as described in Sec. II, and $\boldsymbol{x}^{(1)*}$ and $\boldsymbol{x}^{(N)*}$ are the two extreme points of the Pareto front.

As can be seen in Table 2, for problems 1 and 2, the SM algorithm reduces the drag coefficient value from 203.80 d.c. to 116.9 d.c. and 121.8 d.c, respectively, while satisfying the constraints. Figures 8a and 8b show comparisons of the airfoil shapes and the pressure coefficient distributions of the baseline and SOO optimal designs. Figures 9a–9c show the pressure coefficient contours of the baseline and optimum shape designs, respectively. It can be seen from Figs. 8 and 9 that the optimizer has significantly reduced the strength of the upper surface shock, which explains the large reduction in the drag coefficients.

In terms of the number of model evaluations, SM-based optimization used approximately 500 low-fidelity models and 4 high-fidelity models for both the problems. The cost in terms of central processing unit (CPU) time for the entire optimization process is approximately 14 h on a high-performance computing (HPC) system with 32 processors for each of the SOO problems.

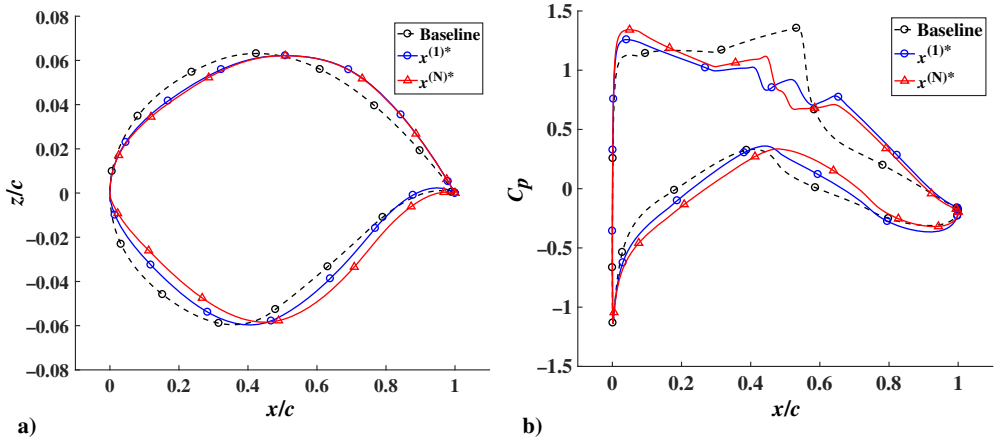

Fig. 8 SOO results showing baseline and optimized a) airfoil shapes and b) pressure distributions at $M_\infty = 0.734$, $C_l = 0.824$, and $Re_\infty = 6.5 \times 10^6$.

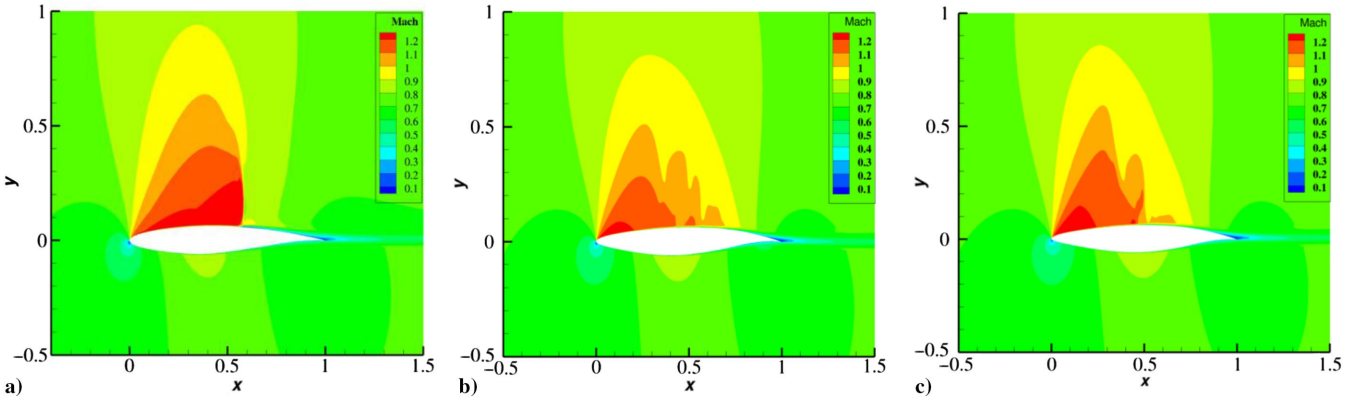

Fig. 9 SOO Mach contours at $M_\infty = 0.734$, $C_l = 0.824$, and $Re_\infty = 6.5 \times 10^6$ of a) the baseline airfoil, b) the SOO optimal design x_1^*, and c) the SOO optimal design x_2^*.

6. Pareto Front

The design space exploration is performed using the algorithm described in Sec. II. Initial designs corresponding to the best possible value of the first objective (minimum drag coefficient) subjected to two different nonlinear pitching moments (target pitching moment values) are obtained in the first step of the process using the output SM-based SOO algorithm [49] as shown in Eqs. (15) and (16). Because of the nature of constraint enforced [cf. Eq. (14)], the MOO algorithm is modified to explore designs outside the design space enclosed within the two SOO points [i.e., points obtained from Eqs. (15) and (16)].

Subsequent designs along the Pareto are obtained using the algorithm on the low-fidelity model as described in Sec. II.C, and the process is terminated when the entire Pareto front is traversed in between the two initial designs. Further, refinement of the initial Pareto front is performed by evaluating the optimal solutions on the high-fidelity model and minimizing the surrogate model constructed at the vicinity of each optimal solutions as discussed in Sec. II.E. The total cost in terms of CPU time to obtain the initial endpoints of the Pareto, the initial Pareto, and the refined Pareto is 27, 13, and 10 h, respectively, on a HPC with 32 processors. Subsequently, the net cost of obtaining the final Pareto front that contains 17 Pareto optimal solutions (as shown in Fig. 10b) is approximately 50 h.

Figure 10a shows the refined optimal solution set (Pareto front) obtained in between the SOO points. A zoomed-in plot of the final Pareto front is represented in Fig. 10b. Few points (point 1 and point 2) on the Pareto optimal set were selected to be compared with the baseline design. Figures 11a and 11b show comparisons of all the airfoil shape designs and the pressure coefficient distributions for

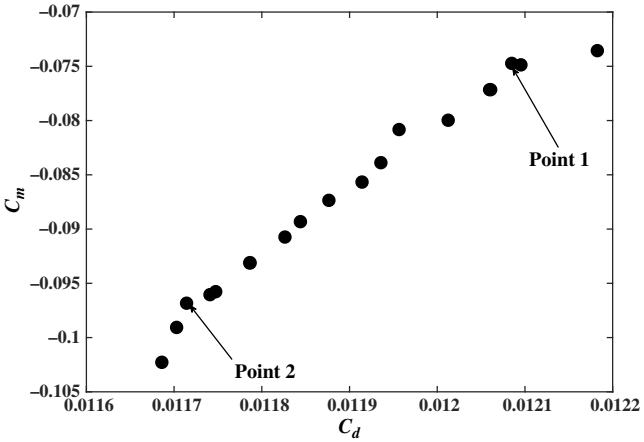

Fig. 10 The final refined Pareto front at $M_\infty = 0.734$, $C_l = 0.824$, and $Re_\infty = 6.5 \times 10^6$.

the selected points and the baseline design. There is a significant difference in the pressure coefficient distribution of the selected points compared with the baseline, with the former having a considerable reduction in shock strength. Further, Mach contour plots in Fig. 12 show point 1 with higher shock strength, leading to more drag compared with point 2. This aligns with the fact that to obtain a lower drag there will be a decrease in pitching moment as shown in Fig. 10. Each CFD simulation is converged to within 1 drag count, and hence irregularity in the Pareto front is observed if zoomed in as in Fig. 10b.

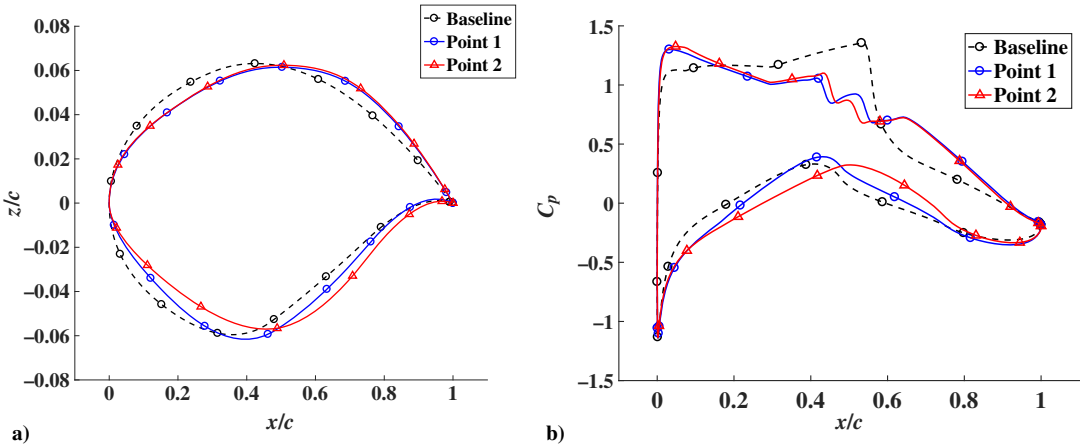

Fig. 11 MOO results showing a) the airfoil shapes and b) the pressure distributions at $M_\infty = 0.734$, $C_l = 0.824$, and $Re_\infty = 6.5 \times 10^6$.

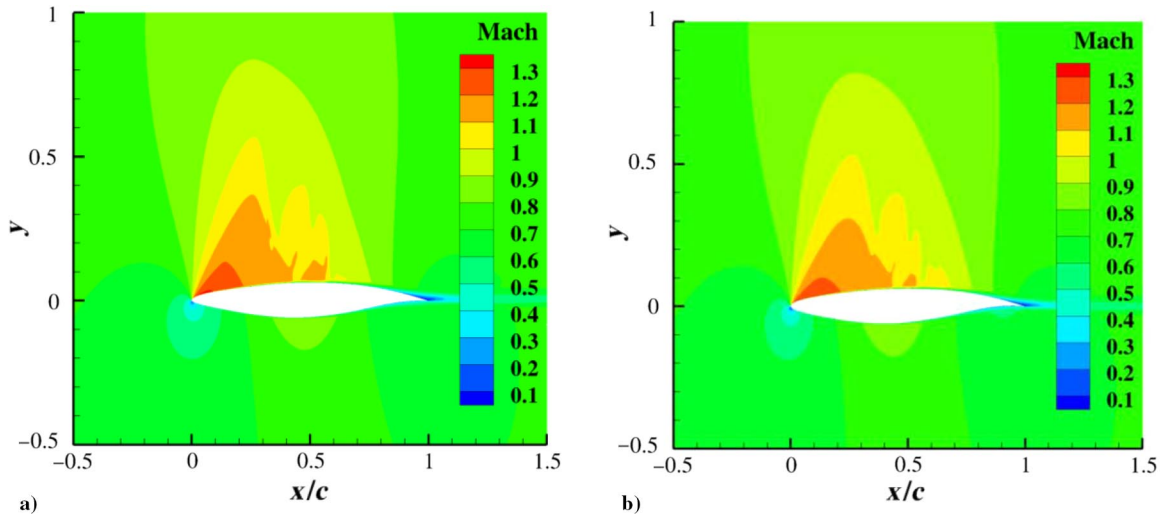

Fig. 12 MOO results showing the Mach contours at $M_\infty = 0.734$, $C_l = 0.824$, and $Re_\infty = 6.5 \times 10^6$ of a) point 1 and b) point 2.

V. Conclusions

This paper presents a unique methodology for the design exploration of aerodynamic problems in a multi-objective form. The extreme points of the Pareto front to be explored are obtained cheaply using multifidelity models and SOO. Further, starting from the extreme points, an approximate Pareto front is identified by constructing patches and using fast low-fidelity aerodynamic models to search for optimum points in those patches. A refinement of the approximate Pareto front is performed using high-fidelity models to obtain the final Pareto front. The key features of the proposed algorithm that distinguishes it from other surrogate-assisted multi-objective aerodynamic design exploration methods are 1) it uses fast low-fidelity models to identify an initial Pareto front quickly, 2) it uses few high-fidelity model evaluations to refine and obtain the final accurate Pareto front, 3) objective function aggregation is not required, and 4) gradient information of the objective function is not used to obtain the Pareto optimal solutions. Future work will investigate the robustness and the scalability of the proposed algorithm.

Acknowledgment

The second and third authors acknowledge the support of the Icelandic Centre for Research (RANNIS) Grant 174573052.

References

[1] Slotnick, J., Khodadoust, A., Alonso, J., Darmofal, D., Gropp, W., Lurie, E., and Mavriplis, D., "CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences," NASA CR-2014-218178, 2014.

[2] Koziel, S., and Leifsson, L., "Surrogate-Based Aerodynamic Shape Optimization by Variable-Resolution Models," *AIAA Journal*, Vol. 51, No. 1, 2013, pp. 94–106. https://doi.org/10.2514/1.J051583

[3] Bandler, J. W., Cheng, Q. S., Dakroury, S., Mohamed, A. S., Bakr, M. H., Madsen, K., and Sondergaard, J., "Space Mapping: The State of the Art," *IEEE Transactions of Microwave Theory and Techniques*, Vol. 52, No. 1, 2004, pp. 337–361. https://doi.org/10.1109/TMTT.2003.820904

[4] Koziel, S., and Bekasiewicz, A., "Multi-Objective Design Optimization of Antenna Structures Using Sequential Domain Patching with Automated Patch Size Determination," *Engineering Optimization*, Vol. 50, No. 2, 2018, pp. 218–234. https://doi.org/10.1080/0305215X.2017.1311879

[5] Queipo, N. V., Haftka, R. T., Shyy, W., Goel, T., Vaidyanathan, R., and Tucker, P. K., "Surrogate-Based Analysis and Optimization," *Progress in Aerospace Sciences*, Vol. 41, No. 1, 2005, pp. 1–28. https://doi.org/10.1016/j.paerosci.2005.02.001

[6] Haftka, R. T., "Combining Global and Local Approximations," *AIAA Journal*, Vol. 29, No. 9, 1991, pp. 1523–1525. https://doi.org/10.2514/3.10768

[7] Rao, S. S., *Engineering Optimization: Theory and Practice*, 3rd ed., Wiley, New York, 1996.

[8] Amrit, A., Leifsson, L., and Koziel, S., "Design Strategies for Multi-Objective Optimization of Aerodynamic Surfaces," *Engineering Computations*, Vol. 34, No. 5, 2017, pp. 1724–1753. https://doi.org/10.1108/EC-07-2016-0239

[9] Obayashi, S., and Sasaki, D., "Finding Tradeoffs by Using Multiobjective Optimization Algorithms," *Transactions of JSASS*, Vol. 47, No. 155, 2004, pp. 51–58.

[10] Buckley, H. P., Zhou, B. Y., and Zing, D. W., "Airfoil Optimization Using Practical Aerodynamic Design Requirements," *Journal of Aircraft*,

Vol. 47, No. 5, 2010, pp. 1707–1719.
https://doi.org/10.2514/1.C000256

[11] Hwang, C. L., and Masud, A. S. M., *Multiple Objective Decision Making, Methods and Applications: A State-of-the-Art Survey*, Lecture Notes in Economics and Mathematical Systems, Springer–Verlag, Berlin, 1979.

[12] Mengistu, T., and Ghaly, W., "Aerodynamic Optimization of Turbomachinery Blades Using Evolutionary Methods and ANN-Based Surrogate Models," *Optimization and Engineering*, Vol. 9, No. 3, 2008, pp. 239–255.
https://doi.org/10.1007/s11081-007-9031-1

[13] Liem, R. P., Martins, J. R. R. A., and Kenway, G. K., "Expected Drag Minimization for Aerodynamic Design Optimization Based on Aircraft Operational Data," *Aerospace Science and Technology*, Vol. 63, April 2017, pp. 344–362.
https://doi.org/10.1016/j.ast.2017.01.006

[14] Zhao, K., Gao, Z., Huang, J., and Li, Q., "Aerodynamic Optimization of Rotor Airfoil Based on Multi-Layer Hierarchical Constraint Method," *Chinese Journal of Aeronautics*, Vol. 29, No. 6, 2016, pp. 1541–1552.
https://doi.org/10.1016/j.cja.2016.09.005

[15] Yang, Z., Cai, X., and Fan, Z., "Epsilon Constrained Method for Constrained Multiobjective Optimization Problems: Some Preliminary Results," *Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation (GECCO Comp '14)*, Assoc. for Computing Machinery, New York, 2014, pp. 1181–1186.

[16] Eiben, A. E., and Smith, J., "From Evolutionary Computation to the Evolution of Things," *Nature*, Vol. 521, No. 7553, 2015, pp. 476–482.
https://doi.org/10.1038/nature14544

[17] Poli, R., Kennedy, J., and Blackwell, T., "Particle Swarm Optimization," *Swarm Intelligence*, Vol. 1, No. 1, 2007, pp. 33–57.
https://doi.org/10.1007/s11721-007-0002-0

[18] Storn, R., and Price, K., "Differential Evolution—A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," *Journal of Global Optimization*, Vol. 11, No. 4, 1997, pp. 341–359.
https://doi.org/10.1023/A:1008202821328

[19] Yang, X. S., "Firefly Algorithms for Multimodal Optimization," *Stochastic Algorithms: Foundations and Applications, SAGA 2009*, Vol. 5792, Lecture Notes in Computer Sciences, Spring–Verlag, Berlin, Germany, 2009, pp. 169–178.

[20] Yang, X. S., and Deb, S., "Cuckoo Search: Recent Advances and Applications," *Neural Computing and Applications*, Vol. 24, No. 1, 2014, pp. 169–174.
https://doi.org/10.1007/s00521-013-1367-1

[21] Forrester, A. I. J., and Keane, A. J., "Recent Advances in Surrogate-Based Optimization," *Progress in Aerospace Sciences*, Vol. 45, Nos. 1–3, 2009, pp. 50–79.
https://doi.org/10.1016/j.paerosci.2008.11.001

[22] Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P., "Design and Analysis of Computer Experiments," *Statistical Science*, Vol. 4, No. 4, 1989, pp. 409–423.
https://doi.org/10.1214/ss/1177012413

[23] Yondo, R., Andres, E., and Valero, E., "A Review on Design of Experiments and Surrogate Models in Aircraft Real-Time and Many-Query Aerodynamic Analyses," *Progress in Aerospace Sciences*, Vol. 96, Jan. 2018, pp. 23–61.
https://doi.org/10.1016/j.paerosci.2017.11.003

[24] Knowles, J., "ParEGO: A Hybrid Algorithm with On-Line Landscape Approximation for Expensive Multiobjective Optimization Problems," *IEEE Transactions on Evolutionary Computation*, Vol. 10, No. 1, 2006, pp. 50–66.
https://doi.org/10.1109/TEVC.2005.851274

[25] Shan, S., and Wang, G., "An Efficient Pareto Set Identification Approach for Multiobjective Optimization on Black-Box Functions," *Journal of Mechanical Design*, Vol. 127, No. 5, 2005, pp. 866–874.
https://doi.org/10.1115/1.1904639

[26] Forrester, A., Sobester, A., and Keane, A., *Engineering Design via Surrogate Modelling: A Practical Guide*, Wiley, New York, 2008.

[27] Praveen, C., and Duvigneau, R., "Low Cost PSO Using Metamodels and Inexact Pre-Evaluation: Application to Aerodynamic Shape Design," *Computer Methods in Applied Mechanics and Engineering*, Vol. 198, Nos. 9–12, 2009, pp. 1087–1096.
https://doi.org/10.1016/j.cma.2008.11.019

[28] Karakasis, M. K., and Giannakoglou, K. C., "On the Use of Metamodel-Assisted, Multi-Objective Evolutionary Algorithms," *Engineering Optimization*, Vol. 38, No. 8, 2006, pp. 941–957.
https://doi.org/10.1080/03052150600848000

[29] Li, M., Li, G., and Azarm, S., "A Kriging Metamodel Assisted Multi-Objective Genetic Algorithm for Design Optimization," *Journal of Mechanical Design*, Vol. 130, No. 3, 2008, Paper 031401.
https://doi.org/10.1115/1.2829879

[30] Hu, W., Li, M., Azarm, S., and Almansoori, A., "Multi-Objective Robust Optimization Under Interval Uncertainty Using Online Approximation and Constraint Cuts," *Journal of Mechanical Design*, Vol. 133, No. 6, 2011, Paper 061002.
https://doi.org/10.1115/1.4003918

[31] Hu, W. W., Saleh, K. H., and Azarm, S. S., "Approximation Assisted Multiobjective Optimization with Combined Global and Local Metamodeling," *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Vol. 3: 38th Design Automation Conference, Parts A and B, ASME, New York, 2012, pp. 753–764.

[32] Zhang, L., Zhang, J., Li, T., and Zhang, Y., "Multiobjective Aerodynamic Optimization Design of High-Speed Train Head Shape," *Journal of Zhejiang University-Science A*, Vol. 18, No. 11, 2017, pp. 841–854.
https://doi.org/10.1631/jzus.A1600764

[33] Wang, W., Mo, R., and Zhang, Y., "Multi-Objective Aerodynamic Optimization Design Method of Compressor Rotor Based on Isight," *Procedia Engineering*, Vol. 15, Dec. 2011, pp. 3699–3703.
https://doi.org/10.1016/j.proeng.2011.08.693

[34] Leusink, D., Alfano, D., and Cinnella, P., "Multi-Fidelity Optimization Strategy for the Industrial Aerodynamic Design of Helicopter Rotor Blades," *Aerospace Science and Technology*, Vol. 42, April 2015, pp. 136–147.
https://doi.org/10.1016/j.ast.2015.01.005

[35] Leifsson, L., Koziel, S., and Tesfahunegn, A. Y., "Multiobjective Aerodynamic Optimization by Variable-Fidelity Models and Response Surface Surrogates," *AIAA Journal*, Vol. 54, No. 2, 2016, pp. 531–541.
https://doi.org/10.2514/1.J054128

[36] Koziel, S., Tesfahunegn, A. Y., and Leifsson, L., "Expedited Constrained Multi-Objective Aerodynamic Shape Optimization by means of Physics-Based Surrogates," *Applied Mathematical Modelling*, Vol. 40, Nos. 15–16, 2016, pp. 7204–7215.
https://doi.org/10.1016/j.apm.2016.03.020

[37] Fincham, J. H. S., and Friswell, M. I., "Aerodynamic Optimization of a Camber Morphing Aerofoil," *Aerospace Science and Technology*, Vol. 43, June 2015, pp. 245–255.
https://doi.org/10.1016/j.ast.2015.02.023

[38] Peherstorfer, B., Willcox, K., and Gunzburger, M., "Survey of Multifidelity Methods in Uncertainty Propagation, Inference, and Optimization," *SIAM Review*, Vol. 60, No. 3, 2018, pp. 550–591.

[39] Han, Z. H., Gortz, S., and Zimmermann, R., "Improving Variable-Fidelity Surrogate Modeling via Gradient-Enhanced Kriging and a Generalized Hybrid Bridge Function," *Aerospace Science and Technology*, Vol. 25, No. 1, 2013, pp. 177–189.
https://doi.org/10.1016/j.ast.2012.01.006

[40] Rathinam, M., and Petzold, L., "A New Look at Proper Orthogonal Decomposition," *SIAM Journal on Numerical Analysis*, Vol. 41, No. 5, 2003, pp. 1893–1925.
https://doi.org/10.1137/S0036142901389049

[41] Rozza, G., Huynh, D. B. P., and Patera, A. T., "Reduced Basis Approximation and a Posteriori Error Estimation for Affinely Parametrized Elliptic Coercive Partial Differential Equations," *Archives of Computational Methods in Engineering*, Vol. 15, No. 3, 2008, pp. 229–275.
https://doi.org/10.1007/s11831-008-9019-9

[42] Scholkopf, B., and Smola, A. J., *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, Cambridge, MA, 2001.

[43] March, A., and Willcox, K., "Constrained Multifidelity Optimization Using Model Calibration," *Structural and Multidisciplinary Optimization*, Vol. 46, No. 1, 2012, pp. 93–109.
https://doi.org/10.1007/s00158-011-0749-1

[44] Keane, A. J., "Cokriging for Robust Design Optimization," *AIAA Journal*, Vol. 50, No. 11, 2012, pp. 2351–2364.
https://doi.org/10.2514/1.J051391

[45] Laurenceau, J., and Sagaut, P., "Building Efficient Response Surfaces of Aerodynamic Functions with Kriging and Cokriging," *AIAA Journal*, Vol. 46, No. 2, 2008, pp. 498–507.
https://doi.org/10.2514/1.32308

[46] Qian, P. Z., and Wu, C. J., "Bayesian Hierarchical Modeling for Integrating Low-Accuracy and High-Accuracy Experiments," *Technometrics*, Vol. 50, No. 2, 2008, pp. 192–204.
https://doi.org/10.1198/004017008000000082

[47] Teckentrup, A. L., Jantsch, P., Webster, C. G., and Gunzburger, M., "A Multilevel Stochastic Collocation Method for Partial Differential Equations with Random Input Data," *SIAM/ASA Journal on Uncertainty Quantification*, Vol. 3, No. 1, 2015, pp. 1046–1074.
https://doi.org/10.1137/140969002

[48] Fonseca, C., "Multiobjective Genetic Algorithms with Applications to Control Engineering Problems," Ph.D. Thesis, Det. of Automatic

Control and Systems Engineering, Univ. of Sheffield, Sheffield, England, U.K., 1995.

[49] Ren, Z., Thelen, A. S., Amrit, A., Du, X., Leifsson, L., Tesfahunegn, Y. A., and Koziel, S., "Application of Multifidelity Optimization Techniques to Benchmark Aerodynamic Design Problems," *54th AIAA Aerospace Sciences Meeting*, AIAA Paper 2016-1542, 2016.

[50] Booker, A. J., Dennis, J. E., Jr., Frank, P. D., Serafini, D. B., Torczon, V., and Trosset, M. W., "A Rigorous Framework for Optimization of Expensive Functions by Surrogates," *Structural Optimization*, Vol. 17, No. 1, 1999, pp. 1–13. https://doi.org/10.1007/BF01197708

[51] Fonzeca, C. M., and Fleming, P. J., "An Overview of Evolutionary Algorithms in Multiobjective Optimization," *Evolutionary Computation*, Vol. 3, No. 1, 1995, pp. 1–16. https://doi.org/10.1162/evco.1995.3.1.1

[52] Economon, T. D., Palacios, F., Copeland, S. R., Lukaczyk, T. W., and Alonso, J. J., "SU2: An Open-Source Suite for Multiphysics Simulation and Design," *AIAA Journal*, Vol. 54, No. 3, 2016, pp. 828–846. https://doi.org/10.2514/1.J053813

[53] Spalart, P. R., and Allmaras, S. R., "A One Equation Turbulence Model for Aerodynamic Flows," *38th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA Paper 1992-0439, Jan. 1992.

[54] Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solution of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes," *AIAA 14th Fluid and Plasma Dynamic Conference*, AIAA Paper 1981-1259, June 1981.

[55] Kinsey, D. W., and Barth, T. J., "Description of a Hyperbolic Grid Generation Procedure for Arbitrary Two-Dimensional Bodies," AFWAL TM 84-191-FIMM, Wright-Patterson Air Force Base Aeronautical Lab., 1984.

[56] Koziel, S., Echeverría-Ciaurri, D., and Leifsson, L., "Surrogate-Based Methods," *Computational Optimization, Methods and Algorithms, Series: Studies in Computational Intelligence*, edited by S. Koziel, and X. S. Yang, Springer–Verlag, Berlin, Germany, 2011, pp. 33–60.