Received 4 December 2018
Revised 31 March 2019
2 July 2019
Accepted 3 July 2019

Applications of surrogate-assisted and multi-fidelity multi-objective optimization algorithms to simulation-based aerodynamic design

Anand Amrit and Leifur Leifsson
Department of Aerospace Engineering, Iowa State University, Ames, Iowa, USA

Abstract

Purpose – The purpose of this work is to apply and compare surrogate-assisted and multi-fidelity, multi-objective optimization (MOO) algorithms to simulation-based aerodynamic design exploration.

Design/methodology/approach – The three algorithms for multi-objective aerodynamic optimization compared in this work are the combination of evolutionary algorithms, design space reduction and surrogate models, the multi-fidelity point-by-point Pareto set identification and the multi-fidelity sequential domain patching (SDP) Pareto set identification. The algorithms are applied to three cases, namely, an analytical test case, the design of transonic airfoil shapes and the design of subsonic wing shapes, and are evaluated based on the resulting best possible trade-offs and the computational overhead.

Findings – The results show that all three algorithms yield comparable best possible trade-offs for all the test cases. For the aerodynamic test cases, the multi-fidelity Pareto set identification algorithms outperform the surrogate-assisted evolutionary algorithm by up to 50 per cent in terms of cost. Furthermore, the point-by-point algorithm is around 27 per cent more efficient than the SDP algorithm.

Originality/value – The novelty of this work includes the first applications of the SDP algorithm to multi-fidelity aerodynamic design exploration, the first comparison of these multi-fidelity MOO algorithms and new results of a complex simulation-based multi-objective aerodynamic design of subsonic wing shapes involving two conflicting criteria, several nonlinear constraints and over ten design variables.

Keywords Multi-objective optimization, Pareto front exploration, Transonic airfoil design, Multi-fidelity methods, Simulation-based aerodynamic design, Subsonic wing design

Paper type Research paper

Nomenclature

A = airfoil cross-sectional area, m^2;
a_∞ = speed of sound, m/s;
b = wing span, m;
C_d = airfoil section drag coefficient, $d/(q_\infty c)$;
C_D = wing drag coefficient, $D/(q_\infty S)$;
C_m = airfoil section pitching moment coefficient, $M/(q_\infty c^2)$;
C_{Mx} = wing pitching moment coefficient, $m/(q_\infty cS)$;
C_l = airfoil section lift coefficient, $l/(q_\infty c)$;
C_L = wing lift coefficient, $L/(q_\infty S)$;
$\mathbf{c}$ = low-fidelity model output;

Engineering Computations
Vol. 37 No. 2, 2020
pp. 430-457
© Emerald Publishing Limited
0264-4401
DOI 10.1108/EC-12-2018-0553

c = chord length, m;
d = airfoil section drag force, N;
D = wing drag force, N;
$\mathbf{d}$ = trust-region radius, m;
$\mathbf{f}$ = high-fidelity model output;
H = objective function value;
$\mathbf{l}$ = lower bound of $\mathbf{x}$;
l = airfoil section lift force, N;
L = wing lift force, N;
M_∞ = Mach number, V_∞/a_∞;
m = pitching moment;
n = number of design variables;
N_c = number of coarse model evaluation;
N_f = number of fine model evaluation;
q_∞ = dynamic pressure, $1/2\rho_\infty V_\infty{}^2$;
$\mathbf{s}$ = surrogate model output;
S = wing planform area, m^2;
$\mathbf{u}$ = upper bound of $\mathbf{x}$;
V_∞ = flow speed, m/s;
V = wing volume, m^3;
$\mathbf{x}$ = design variables, m;
y = coordinate along the wing span, m;
$z_{c/4}$ = vertical coordinates of the quarter-chord location in a wing;
γ = wing twist, degrees;
α = angle of attack, degrees;
ρ_∞ = density, kg/m^3; and
η = wing span stations, $y/(b/2)$.

1. Introduction

The design of aerodynamic surfaces is important for modern engineered systems such as unmanned aerial systems, turbomachinery and aircraft. Physics-based simulations are essential in analyzing and optimizing aerodynamic surfaces, especially for capturing nonlinear system behavior and nonlinear interactions between multiple disciplines (Chiba *et al.*, 2005). The main challenges with using accurate physics-based simulations as part of analysis and optimization include:

- high computational cost (ranging from few hours to days or weeks on high-performance computing clusters);
- repetitive evaluations during the design exploration phase; and
- large numbers of conflicting objectives and constraints, and design variables.

In this paper, three recently developed approaches for obtaining the best possible trade-offs for aerodynamic surfaces are applied for the first time to simulation-based aerodynamic design of a subsonic wing and compared in terms of the resulting best trade-offs and the computational overhead.

Design exploration can be executed in various ways:

- design by engineering experience;
- exhaustive search in design space;

- single-objective optimization (SOO); and
- multi-objective optimization (MOO).

Among all SOO techniques, gradient-based search techniques (Economon *et al.*, 2015) and constrained SOO techniques (Dennis and Torczon, 1994; and Conn *et al.*, 2008) have been widely used for aerodynamic design. SOO methods, however, yield only one optimal solution in a single run.

Designers, typically, prefer to have numerous optimal solutions or the best possible trade-offs between multiple conflicting objectives represented by a Pareto front (Hwang and Md Masud, 1979). Approaches to estimate Pareto fronts include evolutionary algorithms (EAs) (Eiben and Smith, 2015), multi-objective genetic algorithms (MOGAs) (Rao, 1996) and multi-objective EAs (MOEAs) (Amrit *et al.*, 2017a). Other multi-objective approaches include particle swarm optimization (Poli *et al.*, 2007), differential evolution (Storn and Price, 1997), firefly (Yang, 2009) and cuckoo search (Yang and Deb, 2014). All these approaches are metaheuristic methods and require many model evaluations, thus making it impossible to use for aerodynamic design exploration problems.

High-fidelity computational fluid dynamics (CFD) simulations are important to the design of complex engineering systems. Unfortunately, high-fidelity CFD simulations can be computationally expensive and, therefore, it can be impractical to apply them directly in MOO. Surrogate-based optimization (SBO) (Forrester and Keane, 2009; Koziel *et al.*, 2011; Queipo *et al.*, 2005) techniques are widely used to alleviate the computational burden. Various surrogate modeling approaches are utilized for this purpose, such as response surface approximations (Queipo *et al.*, 2005), radial-based function models (Koziel *et al.*, 2011) and kriging interpolation (Koziel *et al.*, 2011). The SBO process has been successfully implemented as part of several MOO algorithms (called surrogate-assisted MOO algorithms). The Pareto-based Efficient Global Optimization (Knowles, 2006) (ParEGO) uses the weighted-sum approach and surrogate modeling for MOO. Similarly, the Pareto set pursuing (PSP) approach (Shan and Wang, 2005) utilizes global surrogate models for MOO. Adaptive sampling and surrogate modeling are combined for MOGAs in Li *et al.* (2008), surrogate-assisted MOO using global and local models is introduced in Hu *et al.* (2012) and finally a global approximation-based MOO for robust design under interval uncertainty is described in Hu *et al.* (2011).

Surrogate-based model construction can, however, involve considerable amount of high-fidelity model evaluations, which can become impractical in some cases. Another mechanism to alleviate the computational burden is to use multi-fidelity methods, which are methods that leverage the computational speed of models of low fidelity and the accuracy of the high-fidelity ones (Peherstorfer *et al.*, 2017). The main elements of multi-fidelity methods are low-fidelity modeling methods and model management strategies (Peherstorfer *et al.*, 2017). Low-fidelity modeling methods include reduced order models [e.g. simplified governing equations (Han *et al.*, 2013), and coarse discretization (Koziel and Leifsson, 2013)], projection-based methods [e.g. proper orthogonal decomposition (Rathinam and Petzold, 2003), reduced basis method (Rozza *et al.*, 2008)] and data-fit methods [e.g. radial basis functions, kriging, and support vector regression (Scholkopf and Smola, 2001)]. Model management strategies include adaptation, fusion and filtering (Peherstorfer *et al.*, 2017). Filtering methods invoke the high-fidelity model following the evaluation of a low-fidelity filter. Examples of filtering methods include the multi-fidelity stochastic collocation approach (Teckentrup *et al.*, 2015). Fusion methods construct a fast multi-fidelity model by combining the output data from the low- and high-fidelity models. Examples of such methods include cokriging (Laurenceau and Sagaut, 2008) and Bayesian regression

(Qian and Wu, 2008). Adaptation approaches are either global methods [e.g. efficient global optimization (EGO) using global data-fit models and infill criteria (Queipo *et al.*, 2005)] or local methods [e.g. space mapping (Bandler *et al.*, 2004), comprehensive (Keane, 2012), multiplicative (Xu *et al.*, 2017) or SBO methods using local data-fit models (Queipo *et al.*, 2005)]. Han and Görtz (2012) and Ariyarit *et al.* (2018) developed several hierarchical kriging surrogate modeling approaches for aerodynamic design applications.

Approaches for multi-objective design exploration of aerodynamic surfaces typically rely on the weighted-sum method and gradient-based search techniques (Nemec *et al.*, 2004; Epstein and Peigin, 2004; Liem *et al.*, 2017). There have been, however, several efforts using evolutionary methods and surrogate models. Multi-objective design optimization of regional jet aircraft wing shapes was presented by Chiba *et al.* (2005) using high-fidelity CFD models and evolutionary algorithms. Mengistu and Ghaly (2008) demonstrated MOO of turbine blades using EAs and surrogate models based on artificial neural networks. Xu *et al.* (2017) demonstrated multi-objective aerodynamic design optimization of high-speed trains using kriging surrogate models and GAs. Shimoyama *et al.* (2012) compared criteria for updating Kriging models in MOO. Emmerich *et al.* (2011) developed hypervolume-based expected improvement criterion. Wang *et al.* (2014), Huang (2014), Huang *et al.* (2013), Ou *et al.* (2019) and Zhang *et al.* (2016a, 2016b) used surrogate modeling and MOO techniques to explore the designs of high-speed aerospace vehicles.

As far is known, not many works involving multi-objective aerodynamic design optimization using multi-fidelity methods have been reported in the literature. Leifsson *et al.* (2016) used multi-fidelity models and local response surface surrogate models for multi-objective design exploration of transonic airfoil shapes. Amrit *et al.* (2017a) developed design strategies for multi-objective aerodynamic design exploration using design space reduction, evolutionary algorithms, cokriging (Laurenceau and Sagaut, 2008; Keane, 2012) and multi-fidelity models. In our recent work (Amrit *et al.*, 2017b), we developed a multi-objective aerodynamic design optimization algorithm using multi-fidelity models and the point-by-point Pareto set identification technique, originally developed for microwave antenna design (Koziel and Bekasiewicz, 2016). The outcome of these work has shown that multi-fidelity methods can significantly reduce the computational overhead of multi-objective aerodynamic design exploration. Ariyarit and Kanazaki (2017) developed a multi-fidelity, multi-objective efficient global optimization algorithm by combining kriging and RBF and applied it to low-speed, multi-objective airfoil design optimization.

This paper presents new applications and comparisons of three recently developed surrogate-assisted and multi-fidelity MOO algorithms. In particular, the algorithms are:

- The surrogate-assisted MOEA (SA-MOEA) developed by Amrit *et al.* (2017a).
- The multi-fidelity multi-objective aerodynamic design exploration using point-by-point Pareto set identification (Amrit *et al.*, 2017b), and
- The multi-fidelity sequential domain patching (SDP) algorithm, originally developed by Koziel and Bekasiewicz (2018), for the design of microwave antennas.

The novelty of this work is:

- The first applications of the SDP algorithm to multi-fidelity aerodynamic MOO.
- New results of simulation-based multi-objective aerodynamic design of subsonic wing shapes involving two conflicting criteria, several nonlinear constraints and over ten design variables.

Furthermore, the algorithms are applied to an analytical test case and the design of transonic airfoil shapes. The paper presents of the results of these application studies and compares the algorithms in terms of the resulting best possible aerodynamic trade-offs, as well as the computational overhead.

Section 2 gives the details of the multi-objective aerodynamic design exploration algorithms. Section 3 presents results of applications of the algorithms to an analytical problem, transonic airfoil design in viscous flow and subsonic rectangular wing design in inviscid flow. The paper ends with conclusion and remarks on future work in Section 4.

2. Methods

This section describes about Pareto front and its global and local exploration methods, and gives the details of the MOO algorithm and the multi-fidelity modeling.

2.1 Definition of the Pareto front

Here, the concept of the Pareto front is explained using a specific example of an aerodynamic design problem. The goal is to find a trade-off 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 denotes as $\mathbf{f}(\mathbf{x}) = [C_{l.f}(\mathbf{x})\ C_{d.f}(\mathbf{x})\ C_{m.f}(\mathbf{x})]^T$, where $\mathbf{x}$ is the $n \times 1$ vector of design variables.

Let $F_k(\mathbf{x})$, $k = 1, \ldots, N_{obj}$, be a kth design objective of interest. 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 a Pareto dominance relation $\prec$, saying that for the two designs $\mathbf{x}$ and $\mathbf{y}$, we have $\mathbf{x} \prec \mathbf{y}$ ($\mathbf{x}$ dominates $\mathbf{y}$) if $F_k(\mathbf{x}) \leq F_k(\mathbf{y})$ for all $k = 1, \ldots, N_{obj}$, and $F_k(\mathbf{x}) < F_k(\mathbf{y})$ for at least one k (Kinsey and Barth, 1984). The goal of the MOO is to find a representation of a so-called 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 $\mathbf{y} \prec \mathbf{x}$.

2.2 Algorithm 1: Pareto front exploration using Surrogate-Assisted Multi-Objective evolutionary algorithm

The SA-MOEA (Amrit *et al.*, 2017a) utilizes the two ends of the Pareto front as its starting point and is obtained using space-mapping-based single-objective optimizations. To expedite the optimization procedure, the algorithm involves utilization of surrogate models constructed from fast low-fidelity model, $\mathbf{c}$, based on coarse-discretization CFD simulations. A design speedup is achieved by performing most of the operations using the data-driven models while using few high-fidelity models to refine the model and yield a Pareto set that is sufficiently accurate.

The steps of the SA-MOEA (Amrit *et al.*, 2017a) are as follows:

- Setup a physics-based surrogate $\mathbf{s}_0$;
- Perform design space reduction using $\mathbf{s}_0$;
- Sample the design space and acquire the surrogate model data with $\mathbf{s}_0$;
- Construct a kriging surrogate $\mathbf{s}_{\mathrm{KR}}$ based on the data from Step 3;
- Obtain an approximate Pareto set representation by optimizing $\mathbf{s}_{\mathrm{KR}}$ using MOEA (Fonseca, 1995);
- Evaluate the high-fidelity model $\mathbf{f}$ along the Pareto front;
- Construct/update the co-kriging surrogate $\mathbf{s}_{\mathrm{CO}}$;

- Update Pareto set by optimizing $\mathbf{s}_{\mathrm{CO}}$ using MOEA (Fonseca, 1995); and
- If termination condition is not satisfied go to Step 6; else END.

In Step 1, instead of using expensive high-fidelity PDE simulations to search a Pareto front, a fast surrogate model is utilized to speed up the process. The surrogate model includes a combination of physics-based and data-driven surrogate models which are clearly described in Bandler *et al.* (2004). Setting up an accurate surrogate model can be expensive on a large design space. Hence, in Step 2, the design space is reduced (in terms of design variable ranges, as well as dimensionality) within which, kriging (Step 4) and co-kriging (Step 7) models are setup using a limited number of model evaluations. The design space reduction methodology is described (Amrit *et al.*, 2017a). In Step 3, Latin Hypercube Sampling (Beachkofski and Grandhi, 2002) (LHS) is used to select the shapes within the reduced design space and the corresponding low-fidelity model values for each sample is collected. A kriging interpolation model is constructed in Step 4, using the sampled data from previous step. Amrit *et al.* (2017a) describes the surrogate model construction. In Step 5, an MOEA is run using the kriging model constructed in Step 4. However, the Pareto optimal set obtained is not accurate, as the kriging model was based on low-fidelity model. In Steps 6 and 7, a few designs are selected uniformly along the Pareto front predicted by the kriging model optimization. A high-fidelity model evaluation is performed on the selected designs and are used to construct a cokriging model as explained by Amrit *et al.* (2017a) and Laurenceau and Sagaut (2008). Step 8 involves using the MOEA and the co-kriging model to refine the Pareto front. If the alignment between the Pareto front and the samples evaluated on high-fidelity 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 Step 9, if the convergence criteria is not met, the cycle is repeated from Step 6 until convergence.

2.3 Algorithm 2: Pareto front exploration using sequential domain patching algorithm

The SDP algorithm involves obtaining optimal solutions in between two points on the Pareto-front. Once the extreme ends of the Pareto front to be explored are obtained [two constrained SOOs are executed (Koziel and Bekasiewicz, 2018)], the SDP-based MOO algorithm is executed, as explained in Koziel and Bekasiewicz (2018), to obtain an initial Pareto set (see Figure 1).

The multi-fidelity SDP algorithm, as shown in Figure 1, produces a sequence of designs, $\mathbf{x}^{(k)*}$, $k = 1, 2, \ldots, N$, where $\mathbf{x}^{(1)*}$ and $\mathbf{x}^{(N)*}$ are the extreme ends of the Pareto front and solutions to the SOO problem. It can be formally summarized as shown below considering only two design objectives:

(1) Patch size $\mathbf{d} = [d_1 \, [\ldots] \, d_n]^T$ is set using the procedure described in Koziel and Bekasiewicz (2018);

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

(3) n perturbations of the size $\mathbf{d}$ are evaluated around $\mathbf{x}_{c1}$ (towards $\mathbf{x}_{cN}$ 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; $\mathbf{x}_{c1}$ is updated;

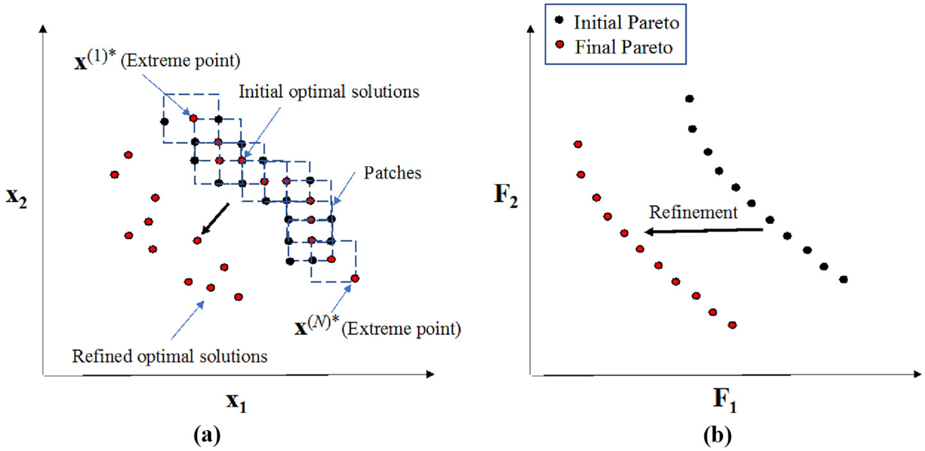

$\mathbf{x}^{(1)*}$ (Extreme point)
Initial optimal solutions
Patches
$\mathbf{x}^{(N)*}$ (Extreme point)
Refined optimal solutions
$\mathbf{x}_2$
$\mathbf{x}_1$
(a)
● Initial Pareto
● Final Pareto
Refinement
$\mathbf{F}_2$
$\mathbf{F}_1$
(b)

Figure 1. An illustration of the SDP algorithm showing (a) the design space and (b) the Pareto front in the feature space

(5) n perturbations of the size $\mathbf{d}$ are evaluated around $\mathbf{x}_{cN}$ (towards $\mathbf{x}_{c1}$ 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; $\mathbf{x}_{cN}$ is updated;

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

The multi-fidelity SDP algorithm discussed above is used to determine the initial Pareto at the level of low-fidelity model $\mathbf{c}$. To obtain the high-fidelity Pareto-optimal designs $\mathbf{x}_\mathrm{c}^{(k)}$, $k = 1,\ldots,N$, the following Pareto set refinement procedure is executed:

$$\mathbf{x}_\mathrm{f}^{(k)} \leftarrow \arg \min_{\mathbf{x}, F_2(\mathbf{x}) \le F_2\left(\mathbf{x}_\mathrm{f}^{(k)}\right)} F_1\left(\mathbf{s}_\mathrm{q}(\mathbf{x}) + \left[\mathbf{f}\left(\mathbf{x}_\mathrm{f}^{(k)}\right) - \mathbf{s}_\mathrm{q}\left(\mathbf{x}_\mathrm{f}^{(k)}\right)\right]\right) \tag{1}$$

In this refinement process, the first objective is improved without degrading the second objective. The above process begins with $\mathbf{x}_\mathrm{f}^{(k)} = \mathbf{x}_\mathrm{c}^{(k)}$ as the starting point and the process is iterated until convergence is met. The correction term in equation (1) makes sure that $\mathbf{s}_\mathrm{q}(\mathbf{x}_\mathrm{f}^{(k)}) = \mathbf{f}(\mathbf{x}_\mathrm{f}^{(k)})$ at the initiation of each iteration. The surrogate model, $\mathbf{s}_\mathrm{q}$, used in this process is a second-order polynomial approximation without the mixed terms given by:

$$\mathbf{s}_\mathrm{q}(\mathbf{x}) = a_0 + \sum_{k=1}^{n} a_k x_k + \sum_{k=1}^{n} a_{n+k} x_k^2, \tag{2}$$

where a_k, $k = 0, 1, 2,\ldots, 2n$, are the coefficients of the quadratic RSA, and x_k, $k = 1, \ldots, n$ is the kth component of the variable vector $\mathbf{x}$. This approximation model is based on low-fidelity model, $\mathbf{c}$ evaluated at $\mathbf{x}_\mathrm{c}^{(k)}$ and the perturbed designs within the patch surrounding $\mathbf{x}_\mathrm{c}^{(k)}$.

The algorithm covers a part of the design space in the form of patches that contains the initial approximation of the set of Pareto-optimal solutions. The total computational cost of the algorithm can be computed as $(M - 1)$ $(n - 1)$ which excludes the cost of solving equation (1), where $M = \sum_{k = 1,...,n} m_k$, and m_k is the number of intervals in the direction j. However, in practice, the total cost can be lower, as some perturbations may not be evaluated owing to the imposed constraints.

In Step 1, two SOO problems are solved on a surrogate model, constructed using the space mapping algorithm (Ren *et al.*, 2016), to obtain the two end points of the Pareto-front to be explored. The solutions from the SOO problems are used as an input to the automated domain patching algorithm [explained in Koziel and Bekasiewicz (2018)] to obtain the patch size **d**. Solutions from the initial two SOO problems are marked in Figure 1(b) as $\mathbf{x}^{(1)*}$ and $\mathbf{x}^{(N)*}$ In Step 2, the next optimum solution is searched in the vicinity of these starting points while moving in either direction, i.e. from $\mathbf{x}^{(1)*}$ to $\mathbf{x}^{(N)*}$ or vice versa. In Step 3, a patch is constructed with n perturbations of size **d** around $\mathbf{x}^{(1)*}$ and/or $\mathbf{x}^{(N)*}$. Each perturbation is evaluated on the low-fidelity model to obtain objective functions and constraints values. The design that brings the largest improvement with respect to the second objective F_2. The search for largest improvement in F_2 is performed with a given condition that the designs are well within the global bounds and also they satisfy the linear and non-linear constraints. To satisfy the constraints, a surrogate model-like equation can be used as discussed in Koziel and Bekasiewicz (2018). In Step 4, the best perturbation result obtained from Step 3 is used to update Step 2, i.e. $\mathbf{x}_{c1}$ is updated. The patch is relocated, so that the center of the patch is the updated $\mathbf{x}_{c1}$. A patch is constructed in Step 5 with n perturbations of size **d** around $\mathbf{x}^{(N)*}$. Each perturbation is evaluated on the low-fidelity model to obtain objective functions and constraints values. The design that brings the largest improvement with respect to the second objective F_1. The search for largest improvement in F_1 is performed such that the designs are well within the global bounds and also they satisfy the linear and non-linear constraints. In Step 6, the best perturbation result obtained from Step 5 is used to update Step 2, i.e. $\mathbf{x}_{cN}$ is updated. The patch is relocated, so that the center of the patch is the updated $\mathbf{x}_{cN}$. Finally, in Step 7, the steps from 2-6 is continued until the path between $\mathbf{x}^{(1)*}$ and $\mathbf{x}^{(N)*}$ is complete.

2.4 Algorithm 3: Pareto front exploration using the point-by-point algorithm

The multi-fidelity point-by-point algorithm (Amrit *et al.*, 2017b) utilizes any single point on the Pareto front as its starting point and is obtained by a space-mapping-based single objective optimization. From the starting point, the MOO algorithm produces a sequence of optimal designs $\mathbf{x}^{(k)}$, $k = 1, 2, \ldots$, point by point. The starting point, $\mathbf{x}^{(1)}$, is a solution to the SOO problem of the form:

$$\mathbf{x}^{(1)} = \arg\min_{\mathbf{x}} F_1\left(\mathbf{f}(\mathbf{x})\right), \tag{3}$$

subjected to:

$$F_2(\mathrm{f}(\mathrm{x})) - b \leq 0,$$

where b is the threshold value for the second objective, and $\mathbf{f}(\mathbf{x})$ is the output of the accurate high-fidelity simulation model. Thus, an optimum design value of the first objective is obtained which lies on the Pareto front.

The SOO process in equation (3) is expedited by a trust region-based multi-fidelity optimization algorithm (Fernandez-Godino *et al.*, 2017; Peherstorfer *et al.*, 2017; Burgee *et al.*, 1996; Alexandrov *et al.*, 2001) using output space mapping (Bandler *et al.*, 2004) to construct the multi-fidelity model. The output space mapping approach exploits a combination of the accurate high-fidelity model **f**, and a model **c**, which is of lower fidelity than **f**, but is computationally faster to evaluate. Here, the low-fidelity model **c** is based on coarse-discretization CFD simulations (Fernandez-Godino *et al.*, 2017; Peherstorfer *et al.*, 2017; Burgee *et al.*, 1996; Alexandrov *et al.*, 2001) for a discussion on approaches for low-fidelity modeling).

Starting from the optimal design obtained from equation (3), further designs that are determined along the Pareto front, $\mathbf{x}^{(k)}$, $k = 2, 3, \ldots$, are obtained by exploring it point by point (Koziel and Bekasiewicz, 2016), using a constrained SOO process as shown below. Let $F_2{}^{(k)}$ be the threshold value for the second objective, then we have:

$$\mathbf{x}^{(k)} = \arg \min_{\mathbf{x}, F_2(\mathbf{f}(\mathbf{x})) \leq F_2{}^{(k)}} F_1(\mathbf{f}(\mathbf{x})). \tag{4}$$

Here, $\mathbf{x}^{(k)}$, $k = 2, 3, \ldots$, is the k^{th} element of the Pareto set and the process is continued until the design specifications are met. Pareto optimal points can be obtained in both directions along the Pareto, i.e. above and below the optimal design from solution of equation (3). The search for the point $\mathbf{x}^{(k)}$ close to the design $\mathbf{x}^{(k-1)}$ is performed using SBO as follows. Specifically, $\mathbf{x}^{(k)}$ is obtained iteratively as a sequence of solutions to (4) $\mathbf{x}^{(k,j)}$, $j = 0, 1, \ldots$, with $\mathbf{x}^{(k,0)} = \mathbf{x}^{(k-1)}$, as follows:

$$\mathbf{x}^{(k,j+1)} = \arg \min_{\mathbf{x}, F_2(\mathbf{s}(\mathbf{x})) \leq F_2{}^{(k)}} F_1\left(\mathbf{s}^{(k,j)}(\mathbf{x})\right), \tag{5}$$

where the surrogate $\mathbf{s}^{(k,j)}$ is defined as:

$$\mathbf{s}^{(k,j)}(\mathbf{x}) = \mathbf{s}_{\mathrm{q}}^{(k,j)}(\mathbf{x}) + \left[\mathbf{f}\left(\mathbf{x}^{(k,j)}\right) - \mathbf{s}_{\mathrm{q}}^{(k,j)}\left(\mathbf{x}^{(k,j)}\right)\right] \tag{6}$$

where $\mathbf{s}_{\mathrm{q}}{}^{(k,j)}$ is a local response surface approximation (RSA) (Koziel *et al.*., 2011) model of **c**, constructed in the vicinity of current design $\mathbf{x}^{(k,j)}$. The vicinity is defined here by the patch (or interval range) $[\mathbf{x}^{(k,j)} - \delta, \mathbf{x}^{(k,j)} + \delta]$, where δ is the trust region radius (Conn *et al.*., 2008). The RSA model is described here below.

The multi-fidelity point-by-point algorithm is summarized as follows:

(1) Obtain $\mathbf{x}^{(1)}$ by performing single-objective multi-fidelity optimization;

(2) Set $F_2{}^{(1)} = F_2(\mathbf{f}(\mathbf{x}^{(1)}))$ and set $k = 2$;

(3) Set $\mathbf{x}^{(k,0)} = \mathbf{x}^{(k\text{-}1)}$;

(4) Evaluate $\mathbf{f}^{(k,j)}(\mathbf{x}^{(k,j)})$;

(5) Construct a local RSA model $\mathbf{s}_{\mathrm{q}}{}^{(k,j)}(\mathbf{x})$ within the interval defined as $[\mathbf{x}^{(k,j)} - \delta, \mathbf{x}^{(k,j)} + \delta]$;

(6) Use (5) to obtain $\mathbf{x}^{(k,j+1)}$ by optimizing the surrogate model $\mathbf{s}^{(k,j)}(\mathbf{x})$;

(7) If $j = 2$, set $\mathbf{x}^{(k)} = \mathbf{x}^{(k,j+1)}$ and go to Step 8; otherwise set the index $j = j + 1$ and go to Step 4;

(8) If the termination condition is met, terminate the algorithm; otherwise update the index $k = k + 1$, set the threshold $F_2{}^{(k)}$ and go to Step 3.

A local patch is formed encompassing the starting point obtained from Step 4, as shown in Figure 2. The patch dimension is determined by performing a sensitivity analysis once and is given by $[\mathbf{x}^{(k,j)} - \delta, \mathbf{x}^{(k,j)} + \delta]$. Normally, δ is set at 0.5-1 per cent of the difference of upper and lower bounds of the design space.

Furthermore, in Step 5, the RSA is constructed within the local patch using a quadratic model $\mathbf{s}_q$ without mixed terms [cf. equation (2)]. In this work, the surrogate $\mathbf{s}_q$ is setup using a star distribution with $2n + 1$ samples (Koziel *et al.*, 2011) evaluated by the low-fidelity model ($\mathbf{c}$). Subsequently, the surrogate model "$\mathbf{s}$" is constructed using the RSA and one high-fidelity model evaluation [cf. equation (6)].

A gradient-based search engine is used to perform the minimization as discussed in Step 6 with the gradients obtained by finite differentiation of the surrogate model [cf. equation (5)]. The target values of the second objective function, $F_2^{(k)}$, is set *a priori* by the designer, and the inequality constraint is handled directly. The results from gradient-based algorithm is checked for the following three criteria, namely, $\|\mathbf{x}^{(i)} - \mathbf{x}^{(i\text{-}1)}\| < \varepsilon_x$, $|F_1^{(i)} - F_1^{(k)}| < \varepsilon_F$ and $\delta^{(i)} < \varepsilon_\delta$, where ε_x, ε_F and ε_δ are user-defined convergence tolerances. In our implementation, the three criteria are combined through the logical alternative (OR). If the criteria are met, it is evaluated on the high-fidelity model, to obtain the target point. If not, the patch is updated using a trust region radius framework (Eiben and Smith, 2015), based on which the trust-region size is reduced by setting $\delta = \delta/m$, where $m = 3$, or increased by setting $\delta = \delta \cdot m$, where $m = 2$.

In Step 7, the target point obtained from Step 6 is set as the starting point for the next algorithm iteration. The process is continued until the designer is satisfied (or a computational budget is exhausted), or the extreme point of the Pareto front is reached (i.e. the first objective cannot be further improved).

3. Numerical examples

This section presents results of the surrogate-assisted and multi-fidelity MOO algorithms on two analytical problems, and the aerodynamic design of transonic airfoils and subsonic wings.

3.1 Analytical problems

3.1.1 The Fonseca and Fleming function. The objective functions of this test problem are given by Booker *et al.* (1999):

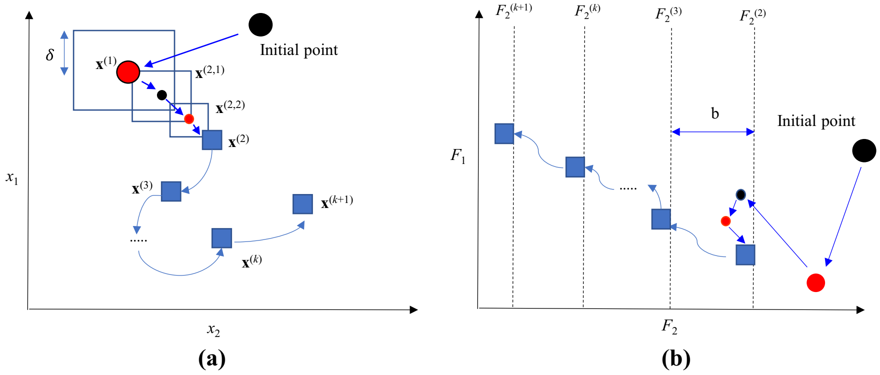


Figure 2. An illustration of the point-by-point Pareto front exploration progression for (a) two design variables in the design space and (b) two objectives in the feature space

$$\min F_1 = 1 - \exp\left[-\sum_{i=1}^{5}\left(x_i - \frac{1}{\sqrt{5}}\right)^2\right], \tag{7}$$

$$\min F_2 = 1 - \exp\left[-\sum_{i=1}^{5}\left(x_i + \frac{1}{\sqrt{5}}\right)^2\right], \tag{8}$$

where the design variables are $x_i \in [-4, 4]$. The analytical functions F_1 and F_2 are considered as the high-fidelity accurate model **f**. A low-fidelity model **c** is formulated by adding noise (ΔF) to the analytical functions as:

$$F_{1.c} = F_1 + \Delta F, \tag{9}$$

$$F_{2.c} = F_2 + \Delta F, \tag{10}$$

where $\Delta F = 0.1x_1 + 0.5$

Figure 3 shows the characteristic features of low- and high-fidelity models. Figure 4(a) and (b) shows the Pareto optimal designs and the final Pareto front, respectively, obtained by the multi-fidelity MOO algorithms. For the purpose of visualization only a part of the Pareto obtained from SDP and point-by-point algorithm is plotted followed by the entire Pareto from the MOEA algorithm. It can be seen that the Pareto front obtained by the algorithms are comparable.

3.1.2 The Zitzler–Deb–Thiele function. The formulation of this test problem is given by Booker *et al.* (1999):

$$\min F_1 = x_1, \tag{11}$$

$$\min F_2 = u\left(1 - \sqrt{x_1/u}\right), \tag{12}$$

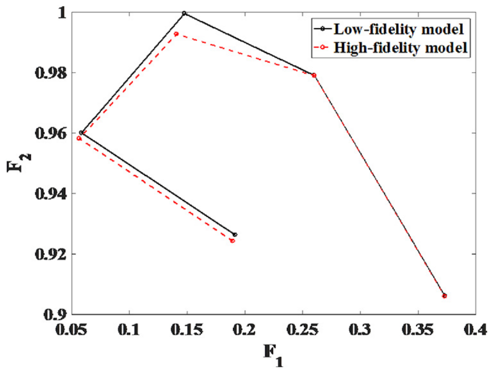


Figure 3.
Example high- and low-fidelity evaluation values of five random designs

where the design variables are $x_i \in [0,1]$, $i = 1, \ldots, 30$ and $u = 1 + 9/7\sum_{i=2}^{30} xi$. As before, the analytical functions F_1 and F_2 are considered as the high-fidelity models, and the low-fidelity model is formulated by adding noise to the analytical functions using equations (9) and (10).

Figure 5 shows the characteristic features of low- and high-fidelity models. Figure 6(a) and (b) shows the Pareto optimal designs and the final Pareto front, respectively, obtained by the multi-fidelity MOO algorithms.

3.2 Transonic airfoil design

In this section, results of the surrogate-assisted and multi-fidelity MOO of transonic airfoil shapes using a viscous fluid flow model is presented. We give the problem description, design variable formulation and results for the SOO and the Pareto front.

3.2.1 Problem description. The main goal of this aerodynamic problem is to obtain the trade-offs between two conflicting objectives, the drag coefficient (C_d) and the pitching moment coefficient (C_m), of the RAE 2822 at a free-stream 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 an area cross-sectional (A) constraint. Specifically, the conflicting objectives considered here are: drag minimization and pitching moment maximization, i.e. we have $F_1(\mathbf{x}) = C_{d.f}$ and

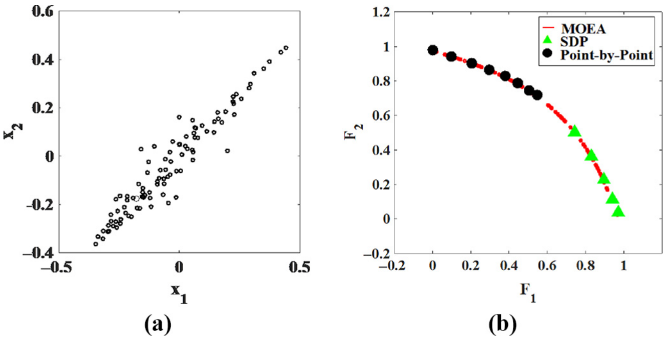


Figure 4. Pareto front obtained for Fonseca and Fleming function (a) design space and (b) feature space

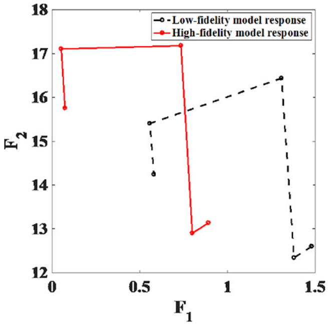
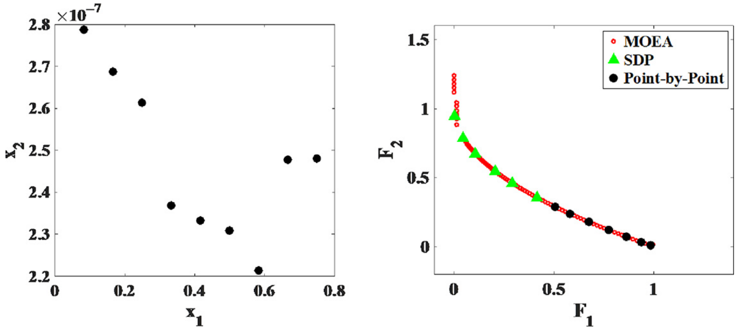


Figure 5. Representation of high- and low-fidelity evaluation values of five random designs



(a) (b)

Figure 6. Pareto front obtained for the ZDT function

Notes: (a) Design space; (b) feature space

$F_2(\mathbf{x}) = C_{m,f}$, and the multi-objective constrained optimization problem can be expressed as:

$$\min_{l \leq x \leq u} C_d, \quad \max_{l \leq x \leq u} C_m, \tag{13}$$

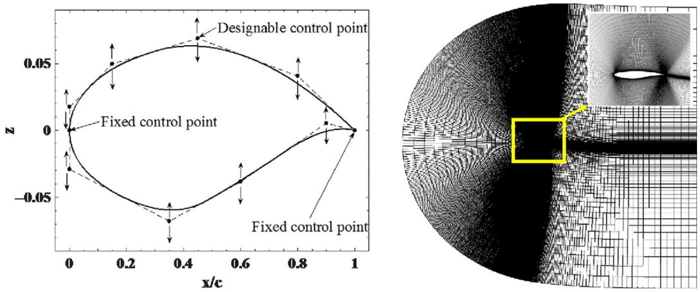
subject to:

$$C_l(\mathrm{x}) = 0.824,$$

$$A(\mathrm{x}) \geq A_{baseline},$$

where $A_{baseline}$ is the cross-sectional area of the baseline RAE2822 airfoil.

3.2.2 Design variables. The airfoil shape is controlled using the B-spline parameterization approach described in Mengistu and Ghaly (2008). Figure 7(a) shows eight control points, four each on the top and bottom surfaces, that can move in the vertical direction. The leading and trailing edge end points of the airfoil are fixed. The design variable vector is $\mathbf{x} = [\mathbf{Z}_u \, \mathbf{Z}_l]^T = [z_{u1} [\ldots] z_{u4} \, z_{l1} [\ldots] z_{l4}]^T$, where z_{ui} and z_{li} are the vertical coordinates of the control points of the upper and lower surfaces, respectively, and $i = 1, [\ldots] 4$, denotes the



(a) (b)

Figure 7. Airfoil models

Notes: (a) Shape parameterization; (b) hyperbolic C-mesh order of the control points from the leading edge toward the trailing edge. The (x/c) locations of the eight control points (eight design variables) are fixed and are based on a fit to the RAE 2822 airfoil shape. Specifically, $\mathbf{X} = [\mathbf{X}_u \ \mathbf{X}_l]^T = [(x/c)_{u1} \, [. . .] \, (x/c)_{u4} \, (x/c)_{l1} \, [. . .] \, (x/c)_{l4}]^T = [0.0 \ 0.15 \ 0.45 \ 0.80 \ 0.0 \ 0.35 \ 0.60 \ 0.90]^T$. The initial design variable vector is $\mathbf{x} = [0.0175 \ 0.0498 \ 0.0688 \ 0.0406 \ -0.0291 \ -0.0679 \ -0.0384 \ 0.0054]^T$. The lower bound of $\mathbf{x}$ is set as $\mathbf{l} = [0.015 \ 0.015 \ 0.015 \ 0.015 \ -0.08 \ -0.08 \ -0.08 \ -0.01]^T$, and the upper bound is set as $\mathbf{u} = [0.08 \ 0.08 \ 0.08 \ 0.08 \ -0.01 \ -0.015 \ -0.015 \ 0.01]^T$.

3.2.3 High-fidelity viscous aerodynamics model. The high-fidelity aerodynamic model (**f**) solves the steady compressible Reynolds-averaged Navier–Stokes (RANS) equations with the Spalart–Allmaras turbulent model (Spalart *et al.*, 1992) using the SU^2 implicit density-based solver (Economon *et al.*, 2016). The second-order JST scheme (Jameson *et al.*, 1981) is used to calculate the convective flux along with one multi-grid 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 (1984) [see Figure 7(b)] with the farfield 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 airfoil surface.

Table I 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. The simulation time presented in Table I is the overall time to compute the constant lift condition using 32 processors. Table I indicates that Mesh 4 is the finest and the most accurate. However, the difference in coefficient of drag between Meshes 3 and 4 is negligible compared to the difference in overall simulation time. Hence, Mesh 3 was chosen as the high-fidelity model (**f**) for this work.

3.2.4 Low-fidelity viscous aerodynamics model. The model set up for low-fidelity model is same as that for high-fidelity model (**f**), with the grid size being far less than that of high-fidelity model. As shown in Table I, we use Mesh 1 for the low-fidelity model **c**. The low-fidelity model convergence criteria are set with the following values:

(1) 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 5,000.

Table I. Grid convergence study for the baseline airfoil shape RAE 2822

Mesh	No. of elements	Lift counts (l.c.)	Drag counts (d.c.)	Simulation time[a] (min)
1	9,836	0.824	324.6	3.1
2	38,876	0.824	221.5	8.8
3	154,556	0.824	204.8	34.0
4	616,316	0.824	203.0	152.6

Notes: [a]Computed on a high-performance cluster with 32 processors; flow solution only

3.2.5 Single-objective optimization (SOO) results. 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 each algorithm using the output SM-based SOO algorithm (Ren *et al.*, 2016) shown in Table II as SOO 1 and SOO 2. While MOEA and SDP use both the SOO points as their starting points, point-by-point algorithm uses only SOO 1 point. The runs are performed at $M_\infty = 0.734$, $A_{\text{baseline}} = 0.0779$ and at $C_l = 0.824$.

3.2.6 The Pareto front. The design space exploration is performed using all the algorithms described in Section 2. Owing to the computational expense of the CFD models, instead of exploring the entire Pareto, i.e. in between the optimal points of the two objective functions (the extreme points), only a part of the Pareto front is explored. Subsequent designs along the Pareto were obtained using the algorithms described in Section 2, and the process is terminated when the Pareto front is traversed in between the two initial designs. The algorithm is run to obtain a Pareto in between $C_m = -0.074$ and $C_m = -0.11$ (selected here for illustration purposes).

Figure 8 (a) shows the final optimal solution set (Pareto front) obtained in between the SOO points. A zoomed-in plot of the final Pareto front is represented in Figure 8(b). Two points designated as Point 1 and Point 2 on the Pareto optimal set were selected to be compared with the baseline design. Figure 9(a) and (b) shows comparisons of all the airfoil-shape designs and the pressure coefficient distributions for the selected points and the baseline design. There is a significant difference in the pressure coefficient distribution of the selected points compared to the baseline, with the former having a considerable

Table II.
Single objective optimization results for transonic airfoil design case

Parameter/method	Baseline	SOO 1	SOO 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

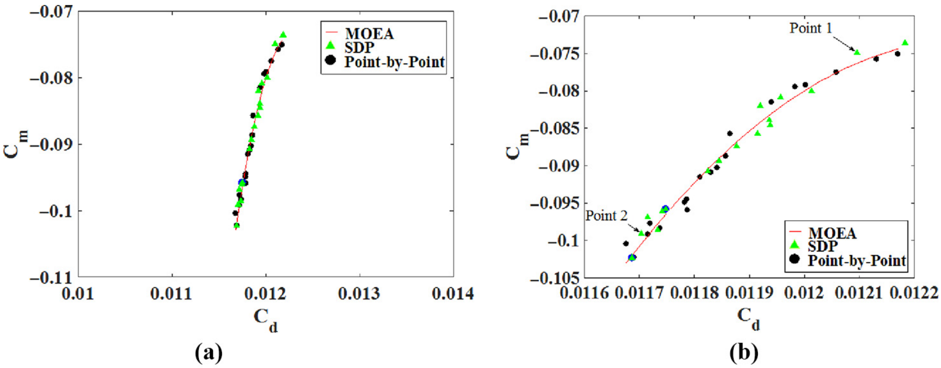
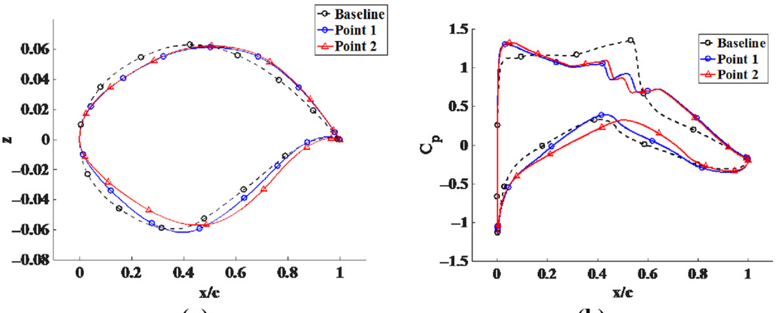


Figure 8.
Final Pareto fronts from all three algorithms at $M_\infty = 0.734$, $C_l = 0.824$ and $Re_\infty = 6.5 \times 10^6$

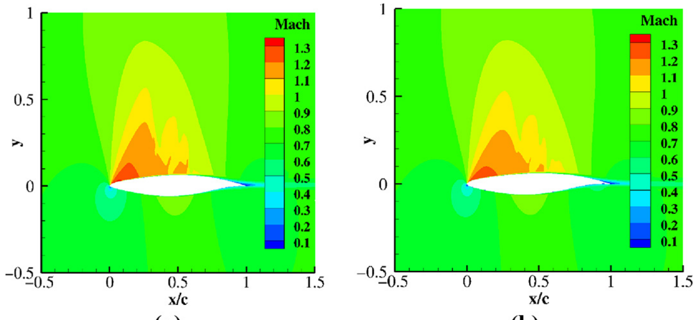
Notes: (a) Final refined Pareto; (b) zoomed-in view of the same Pareto front reduction in shock strength. Further, the Mach contour plots in Figure 10 show Point 1 with higher shock strength, leading to more drag compared to Point 2. This aligns with the fact that to obtain a lower drag, there will be a decrease in pitching moment as shown in Figure 8. Note that the main reason behind irregularity in the Pareto front obtained from SDP and point-by-point algorithms (as shown in Figure 8) is that each CFD simulation is converged to within 1 drag count, and the terminating conditions for reaching a target point in both the algorithms is set to 1 drag count. All the three Pareto fronts are within 1 drag count error.

The total cost in terms of CPU time to obtain the final Pareto on a HPC with 32 processors is shown in Table III, which includes the cost of obtaining the starting points. The MOEA utilizes a fixed number of high-fidelity samples every iteration until convergence for Pareto-front refinement. For transonic airfoil design exploration, the number of MOO iterations need to be increased as compared to simple analytical



(a) (b)

Figure 9. MOO results showing (a) the airfoil shapes and (b) the pressure distributions at $M_\infty = 0.734$, $C_l = 0.824$ and $Re_\infty = 6.5 \times 10^6$ for different points along the Pareto



(a) (b)

Figure 10. Mach contours at $M_\infty = 0.734$, $C_l = 0.824$, $Re_\infty = 6.5 \times 10^6$ of (a) Point 1 and (b) Point 2

Table III. Cost of each multi-fidelity MOO algorithm for the transonic airfoil design case

Algorithm	Name	N_c	N_f	Total time (h)
1	SA-MOEA	50	50	60
2	SDP	350	30	50
3	Point-by-Point	462	22	35

problems (shown in Section 3.1) to explore the nonlinear design space accurately. Owing to this, the MOEA-based MOO approach requires many high-fidelity model evaluations (50 for this case) when compared to the other two algorithms. Apart from that, the MOEA-based algorithm requires the two end points of the Pareto front to be explored and, hence, needs to solve two SOO problems. Thus, the high cost of obtaining the starting points and more dependency on the high-fidelity model evaluations for the refinement step renders the MOEA-based algorithm to be the more expensive than the other two algorithms. The full Pareto front, however, is produced by the MOEA-based algorithm.

The SDP algorithm requires the two end points of the Pareto front to be explored as its starting points as the MOEA-based algorithm. Unlike MOEA, however, the SDP algorithm mostly relies on the low-fidelity model (350 low-fidelity model evaluations) to obtain an initial Pareto front. Furthermore, two refinement steps require 30 high-fidelity model evaluations to obtain 20 optimal points along the Pareto front. The total computational time to obtain the entire Pareto was approximately 50 h.

The point-by-point algorithm is found to be the cheapest among the three algorithms, requiring only 35 h of computational time (22 high-fidelity model evaluations) to obtain 20 Pareto optimal solutions. The high exploitation of the low-fidelity model (462 low-fidelity model evaluations) and the utilization of only one starting point are the reasons for it to be the most efficient algorithm.

3.3 Subsonic rectangular wing design

This section presents results of applications of the surrogate-assisted and multi-fidelity algorithms to the MOO of a rectangular subsonic wing shapes. We give the problem formulation, design variable formulation and results for the SOO and the Pareto front.

3.3.1 Problem description. The objective is to obtain the best possible trade-offs between the drag coefficient (C_D) and the pitching moment coefficient (C_{Mx}) of a rectangular wing (see Figure 11) in an inviscid subsonic flow at a Mach number of $M_\infty = 0.5$ and a fixed lift coefficient of 0.2625 (C_L). The multi-objective constrained optimization problem is written as:

$$\min_{l \le x \le u} C_D, \min_{l \le x \le u} C_M \tag{14}$$

subject to:

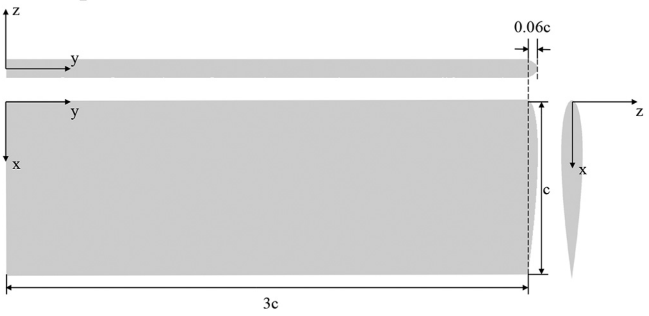


Figure 11. Baseline semi-span wing geometry with a constant section profile

$$C_L = 0.2625,$$

$$100\left(\frac{|S_0 - S|}{S_0}\right) - 1\% \leq 0,$$

$$100\left(\frac{V_0 - V}{V_0}\right) \leq 0,$$

$$-3^\circ - \alpha \leq 0,$$

and:

$$\alpha - 6^o \leq 0.$$

The designable parameters are the wing semi-span ($b/2$), as well as the span-wise distributions of twist (γ), chord (c) and dihedral ($z_{c/4}$). Although the distributions of sweep and airfoil shape can change in the problem, they are fixed to zero center-chord sweep and the NACA 0012 airfoil shape for this case. Six nonlinear constraints are enforced (one equality and five inequality). The equality constraint ensures a lift coefficient of 0.2625, and is enforced implicitly within the flow solver. The first inequality constraint ensures that the planform area remains within 1 per cent of the baseline value of 3.06, while the second ensures that the internal wing volume is greater than or equal to that of the baseline. Lastly, the angle of attack is limited to a range of $-3°$ to $6°$. Here, we represent the volume, area and moment constraints as percentages of the reference values, which, respectively, are the minimum volume ($V_0 = 0.24818$), target planform area ($S_0 = 3.04110$) and the maximum moment coefficient value ($C_{M_{max}} = 0.1069$). The angle of attack is denoted by α.

3.3.2 Design variables. B-spline parameterization (Amrit *et al.*, 2017b) is used to control the surface of the rectangular wing. There are 11 designable parameters: the wing semi-span ($b/2$), and distributions of the wing chord length (c), $z_{c/4}$ (vertical) coordinates at the quarter-chord and the twist (γ) as shown in Figure 12. Specifically, the designable vector is written as $\mathbf{x} = [b/2\ \mathbf{c}\ \mathbf{z}_{c/4}\ \boldsymbol{\gamma}]^T = [b/2\ c_r\ c_{c,y}\ c_{c,z}\ c_t\ (z_{c/4})_{c,y}\ (z_{c/4})_{c,z}\ (z_{c/4})_t\ \gamma_{c,y}\ \gamma_{c,z}\ \gamma_t]^T$, where the subscripts r, c, y, z and t denote the root control point, center control point, horizontal direction, vertical direction and the tip control point. The horizontal coordinates of each center control point can vary between 0.2 and 0.8. For chord, the vertical coordinates of the root and tip control points can vary between 0.45 and 1.55, while the center point can vary

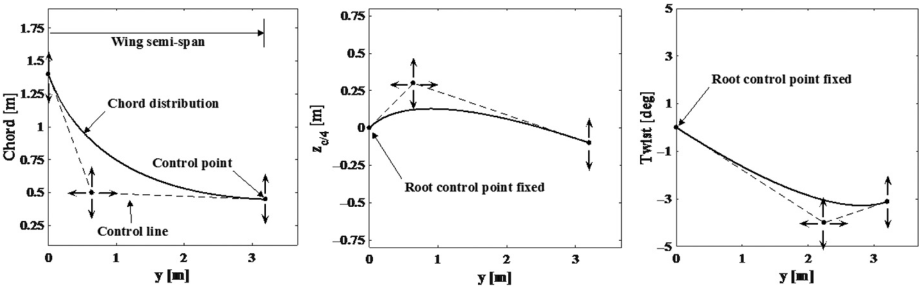


Figure 12. Wing-shape parameterization using B-spline curves for chord length, wing quarter-chord height and twist

between 0.1 and 1.9. For z-coordinates, the tip control point can vary between -0.45 and 0.45 vertically, while this range for the center point is -0.8 to 0.8. Finally, for twist, the tip control point can vary between -3.12 and 3.12 vertically, while the center point can vary from -5 to 5. Nonlinear constraints are enforced to ensure that each of these distributions fall within the ranges [0.45 1.55], [-0.45 0.45] and [-3.12 3.12] for chord, z-coordinates and twist angle, respectively. The initial design variable vector is set $\mathbf{x} = [3.06\ 0.5\ 1\ 1\ 1\ 0.5\ 0\ 0\ 0.5\ 0\ 0]^T$. The lower bound of $\mathbf{x}$ is set as $\mathbf{l} = [2.46\ 0.20\ 0.45\ 0.10\ 0.45\ 0.20\ -0.80\ -0.45\ 0.20\ -5.00\ -3.12]^T$, and the upper bound is set as $\mathbf{u} = [3.67\ 0.80\ 1.55\ 1.90\ 1.55\ 0.80\ 0.80\ 0.45\ 0.80\ 5.00\ 3.12]^T$.

3.3.3 Computational model. To evaluate the loads on a particular wing design, the aerodynamics model executes several sequential steps. First, the design variables are converted to the span-wise geometric distributions (e.g. twist, chord, sectional shape) using B-spline parameterization. Then, using the resulting geometry, an Engineering Sketch Pad (ESP) (Haimes and Dannenhoffer, 2013) script is executed to produce the solid model. This CAD model is then used by a Pointwise script to generate a structured mesh with an O-type topology as shown in Figure 13. Then, SU2 version 5.0.0 (Economon *et al.*, 2016) is used to solve the Euler equations on the grid at the specified Mach number. Each design evaluation adjusts the angle of attack to produce the prescribed lift coefficient implicitly within the flow solver.

Before initiating the optimization process, a grid convergence study was carried out for the baseline design, as shown in Figure 14(a). A set of refinement factors, ranging from 0.25 to 1.75, in increments of 0.25, was chosen to increase the mesh size in the grid convergence study. Each refinement factor value is utilized to divide the off-wall spacing while multiplying the numbers of cells (along wing, around airfoil, between wing and far-field). The resulting meshes had numbers of cells ranging from approximately 25,000 to 9 million cells as shown in Figure 14(a). A low-fidelity model, i.e. a coarse discretized mesh, is selected having an off-wall spacing of 0.004, 100 cells along the span, approximately 170 cells around the airfoil sections and 100 cells between the wing and far-field surface, which is located at a radius of 50 chord lengths. The low-fidelity model with approximately 1.7 M cells and the high-fidelity model with approximately 9 million cells consume around 0.5 h and 2.5 h, respectively, per simulation with 32 processors on a high-performance computing cluster as shown in Figure 14(b).

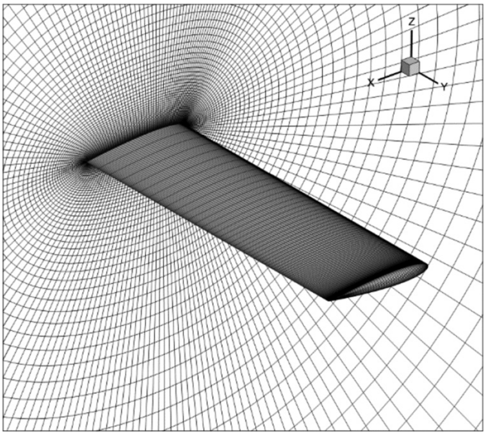


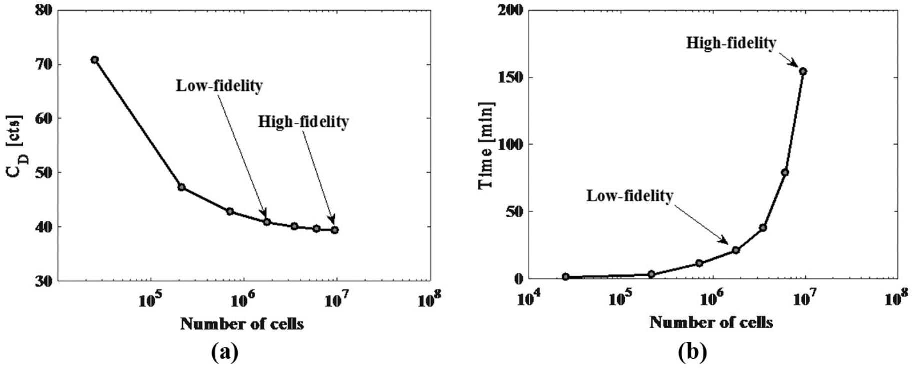
Figure 13.
Sample mesh used for the CFD simulations

3.3.4 Single-objective optimization (SOO) results. The two SOO problems, i.e. to minimize the drag coefficient (C_D) and to minimize the pitching moment coefficient (C_M), are solved using the SM algorithm (Amrit *et al.*, 2017b), and results are shown in Table IV. The cost in terms of CPU time for each of the SOO process is approximately 9 h on an HPC with 32 processors. It should be noted here that while the MOEA-based and SDP algorithms both use the SOO points as their starting points, the point-by-point algorithm uses only SOO 1 point as its starting point.

3.3.5 Pareto front. The Pareto front for the multi-objective aerodynamic design problem is explored by executing all three multi-fidelity algorithms. A target Pareto front is identified to be explored and is determined to be ±10 per cent of the pitching moment value of SOO point (shown in Table IV). Subsequent Pareto optimal designs are obtained using each algorithm discussed in Section 2 and are iterated until their respective terminating conditions are met.

Figure 15(a) shows a comparison of Pareto fronts obtained by executing the algorithms. It can be seen that the Pareto fronts are comparable. Figure 15(b) is a zoomed-in view of the same Pareto with all the three Pareto fronts within less than one drag count (which is within the numerical accuracy level of the high-fidelity model). Two designs, designated as Point 1 and Point 2 on Figure 15(b), are selected from the Pareto for comparison purposes.

Figure 16 shows the design variable distributions for the baseline and the designated points from the Pareto fronts. It indicates a relatively longer chord length along the wing



Notes: (a) Drag coefficient versus number of mesh cells; (b) simulation run-time

Figure 14. Grid convergence results for the baseline design at $M_\infty = 0.5$ and $C_L = 0.2625$

Parameter/Method	Baseline	SOO 1	SOO2
C_D (d.c.)	40.78	32.14	42.41
C_{Mx}	0.11369	0.1194	0.0990
N_f	–	12	12
N_c	–	18,403	18,000
t_c (h)	–	2.95	2.8
t_f (h)	–	6.46	6.4
t_{tot} (h)	–	9.41	9.2
Termination criterion	–	Step size	Step size

Table IV. SOO results for the subsonic wing design

semi-span and higher washout for Point 1 compared to Point 2. Table V gives the numerical values of the wing design parameters for the baseline and both the selected points from the Pareto front at non-dimensionalized wing semi-span locations $\eta = 0.02, 0.5, 0.75$ and 1 (here, $\eta \equiv y/(b/2)$). The results indicate that both Points 1 and 2 have a larger wing semi-span than the baseline, with Point 2 having the largest wing semi-span of 3.40 m. Point 1 has a larger washout than the baseline and Point 2. The maximum washout of Point 1 is −3.12°, whereas Point 1 has a maximum of −1.01°.

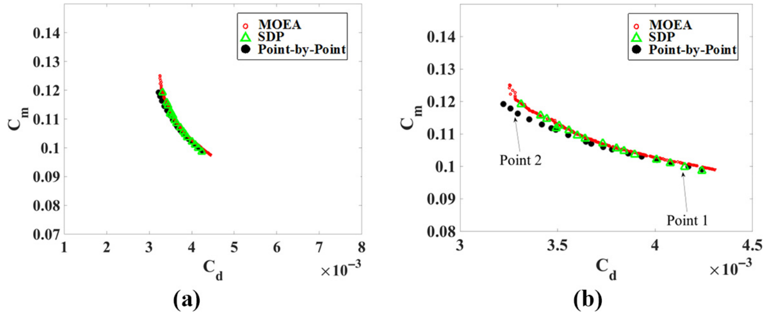


Figure 15. Final Pareto front obtained from each algorithm

Notes: (a) Optimum solutions; (b) zoomed-in plot

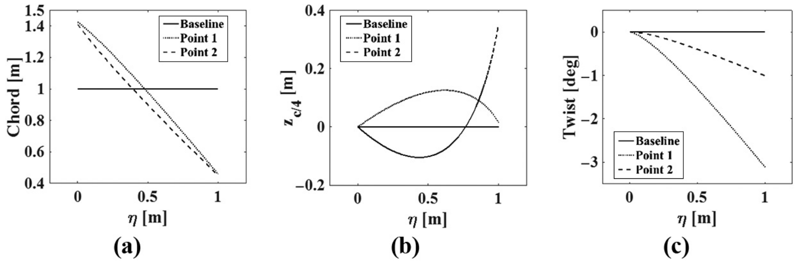


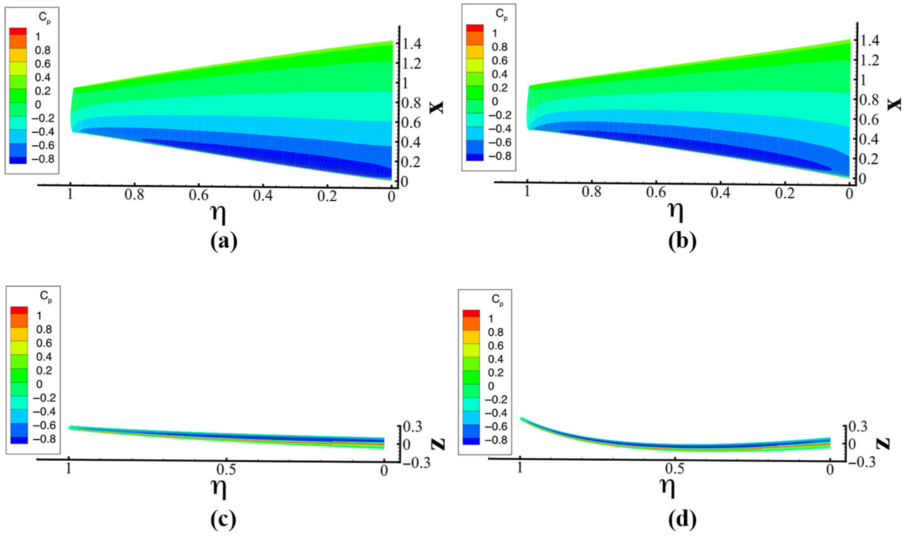
Figure 16. Wing-shape parameter comparison for baseline, Point 1 and Point 2 designs for (a) chord length, (b) wing quarter-chord height and (c) twist.

Table V. Wing-shape parameter comparison for baseline, Point 1 and Point 2 designs at multiple wing span stations

Designs	η	$b/2$ [m]	c [m]	$z_{c/4}$ [m]	γ [deg]
Baseline	0.02	3.06	1	0	0
Point 1	0.02	3.14	1.41	0.01	−0.02
Point 2	0.02	3.40	1.40	−0.01	0
Baseline	0.5	3.06	1	0	0
Point 1	0.5	3.14	0.97	0.12	−1.33
Point 2	0.5	3.40	0.89	−0.10	−0.40
Baseline	0.75	3.06	1	0	0
Point 1	0.75	3.14	0.73	0.12	−2.18
Point 2	0.75	3.40	0.67	−0.01	−0.69
Baseline	1	3.06	1	0	0
Point 1	1	3.14	0.46	0.02	−3.12
Point 2	1	3.40	0.45	0.35	−1.01

A comparison of the wing shapes and surface pressure contours of the designs of Point 1 and Point 2 is shown in Figure 17. Figure 17(a) and (b) shows the planform view, and Figure 17(c) and (d) shows the side view. It can be seen that the tip of the wing for the design designed by Point 2 curves upward, which is a shape similar to a winglet and has the function of reducing lift-induced drag. The pressure coefficient distribution of the designated points along the Pareto is plotted along four wing stations as shown in Figure 18. It can be observed in Figure 18(a) and (b) that although design Point 1 and Point 2 have almost the same root chord length, Point 1 has lower pressure coefficient value at the leading edge owing to a positive dihedral angle, whereas Point 2 has an anhedral angle (see Table V). At η = 0.75, although Point 1 has a higher positive dihedral angle, the pressure coefficient distributions of both points are comparable as shown in Figure 18(c). This is because of the relatively lower twist value for Point 1 which compensates the pressure effects. Further, Figure 18(d) shows that at η = 1, Point 2 has lower pressure coefficient value at the leading edge than Point 1. This is due to a relatively higher dihedral angle for Point 2 and much higher negative twist for Point 1 (cf. Table V). Figure 19 shows a comparison of the wing section lift coefficient distribution of Points 1 and 2. Point 1 has a higher inboard section lift coefficient than Point 2, whereas the outboard section lift coefficients are comparable. Point 2 has a larger semi-span than Point 1 (a difference of 0.26 m or a relative difference of 8.3 per cent). Because the chord lengths of the wings are comparable, the higher pitching moment coefficient of Point 2 is therefore mainly due to the larger wing span.

The best possible trade-offs between the drag coefficient and the pitching moment coefficient for this design problem can be characterized by the wing span and the wing shape and twist near the wing tip. In particular, adding length to the wing span and increasing the curvature of the wing tip with a moderate amount of twist will yield design of a low drag coefficient and a high pitching moment coefficient, whereas reducing the wing



(a)

(b)

(c)

(d)

Notes: (a) Point 1 (top view); (b) Point 2 (top view); (c) Point 1 (side view); (d) Point 2 (side view)

Figure 17. Pressure contours at $M_\infty = 0.5$ and $C_L = 0.2625$

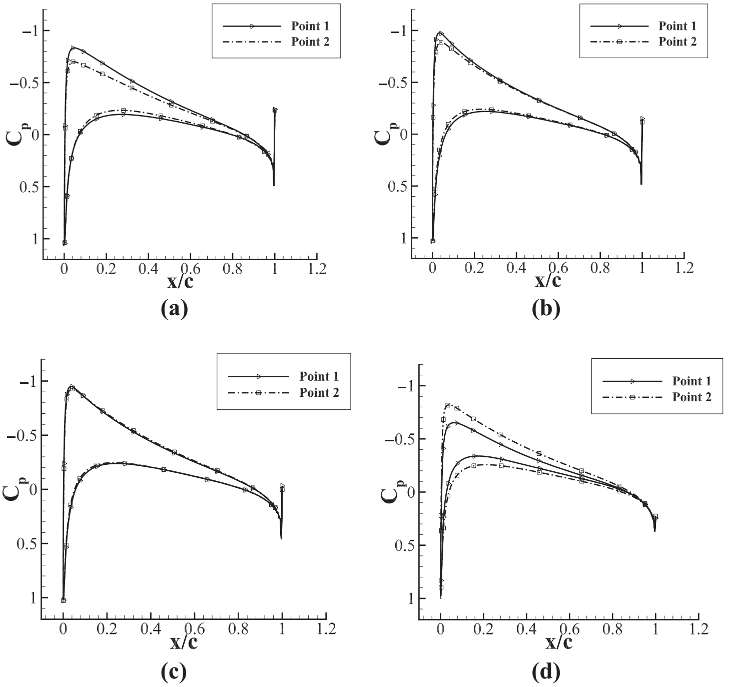


Figure 18. MOO results showing the pressure distributions of two points on the Pareto at $M_\infty = 0.5$ and $C_L = 0.2625$ and at (a) $\eta = 0.02$, (b) $\eta = 0.5$, (c) $\eta = 0.75$ and (d) $\eta = 1$

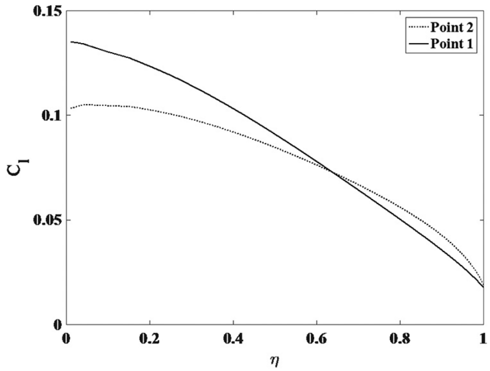


Figure 19. Comparison of the wing lift distributions for the Point 1 and Point 2 designs

span and wing tip curvature and increasing the tip twist will yield designs of a high drag coefficient and a low pitching moment coefficient. For these designs, the wing chord distribution is trapezoidal.

A comparison of the number of low-fidelity model evaluations (N_c), the number of high-fidelity model evaluations (N_f) and the total computational cost of each algorithm to obtain 20 optimal solutions is reflected in Table VI. For this problem, the point-by-point algorithm is the fastest with approximately 40 h of computational time (500 low-fidelity and 30 high-fidelity model evaluations), and the MOEA-based algorithm is the most expensive with approximately 80 h of computational time (50 low-fidelity and 60 high-fidelity model evaluations) on a high-performance computing cluster with 32 processors. In other words, the point-by-point requires around 50 per cent less time than the MOEA-based algorithm. Furthermore, the point-by-point algorithm is around 27 per cent more efficient than the SDP algorithm for this case. The reasons for the point-by-point algorithm being more efficient are, as in transonic airfoil design case, that it requires only one SOO point and exploits the low-fidelity model to obtain the Pareto front at the high-fidelity model level, whereas the MOEA-based and SDP algorithm both require two SOO points.

4. Conclusion

This paper presents the results of applications of three recently developed MOO algorithms to simulation-based aerodynamic design of transonic airfoils and subsonic wings. The surrogate-assisted multi-objective evolutionary algorithm (SA-MOEA) utilizes a combination of design space reduction, surrogate modeling methods and an evolutionary search algorithm. The other two algorithms use multi-fidelity methods and Pareto set identification techniques, namely, the SDP and the point-by-point method, as well as local search with gradient-based techniques. The results of applications show that the algorithms obtain comparable Pareto front solutions. In terms of computational cost, the multi-fidelity point-by-point algorithm outperforms both the SA-MOEA and the multi-fidelity SDP algorithm for the cases considered. Additionally, in the course of this work, it was found that the accuracy and speed of the multi-fidelity algorithms depend significantly on the quality of the low-fidelity model. Specifically, the low-fidelity models need to correlate well with the high-fidelity model. Therefore, a careful selection of the low-fidelity model, such as through a grid independence study, is critical for the successful application of the algorithms.

A distinct advantage of the SA-MOEA algorithm over the other two is that it can yield the entire Pareto front in one algorithm run. This desirable characteristic, however, comes at an additional computational cost. The advantages of the multi-fidelity SDP and point-by-point algorithms over SA-MOEA is that a part of the Pareto front can be estimated based on the available computational budget. The higher computational efficiency of the multi-fidelity point-by-point algorithm is due to the fact that it requires only one starting point on the Pareto front, whereas the SA-MOEA and multi-fidelity SDP algorithm both require two starting points. Furthermore, the multi-fidelity point-by-point and SDP algorithms are more efficient than the SA-MOEA algorithm owing to their capability of exploiting fast low-fidelity models. It should be noted that the overall computational cost of the multi-fidelity point-by-point and SDP algorithms is, however, limited by the efficiency of the search algorithm used to determine the initial points on the Pareto front.

Future work will investigate how the computational cost grows with the complexity of the aerodynamic design problem with respect to the dimensionality of the problem and the degrees of freedom of the simulations. Furthermore, future work will investigate the robustness and reliability of the multi-fidelity point-by-point and SDP algorithms with respect to the setup of the low-fidelity models. Moreover, extensions of the multi-fidelity

Table VI. Computational cost of each MOO algorithm for the subsonic wing design problem

Algorithm	Name	N_c	N_f	Total time (h)
1	SA-MOEA	50	60	80
2	SDP	375	40	55
3	Point-by-Point	500	30	40

point-by-point and SDP algorithms for aerodynamic design exploration with more than two objectives will be considered. Lastly, future work should investigate how the performance of the multi-fidelity algorithms is affected by the selection of the low-fidelity models.

References

Alexandrov, N.M., Lewis, R.M., Gumbert, C.R., Green, L.L. and Newman, P.A. (2001), "Approximation and model management in aerodynamic optimization with variable-fidelity models", *Journal of Aircraft*, Vol. 38 No. 6, pp. 1093-1101.

Amrit, A., Leifsson, L. and Koziel, S. (2017a), "Design strategies for multi-objective optimization of aerodynamic surfaces", *Engineering Computations*, Vol. 34 No. 5, pp. 1724-1753.

Amrit, A., Leifsson, L. and Koziel, S. (2017b), "Multi-fidelity aerodynamic design trade-off exploration using point-by-point pareto set identification", *Aerospace Science and Technology*, Vol. 79, pp. 399-412.

Ariyarit, A. and Kanazaki, M. (2017), "Multi-fidelity multi-objective efficient global optimization applied to airfoil design problems", *Applied Sciences*, Vol. 7 No. 12, pp. 1318.

Ariyarit, A., Sugiura, M., Tanabe, Y. and Kanazaki, M. (2018), "Hybrid surrogate-model-based multi-fidelity efficient global optimization applied to helicopter blade design", *Engineering Optimization*, Vol. 50 No. 6, pp. 1016-1040.

Bandler, J.W., Cheng, Q.S., Dakroury, S., Mohamed, A.S., Bakr, M.H., Madsen, K. and Sondergaard, J. (2004), "Space mapping: the state of the art", *IEEE Transactions on Microwave Theory and Techniques*, Vol. 52 No. 1, pp. 337-361.

Beachkofski, B. and Grandhi, R. (2002), "Improved distributed hypercube sampling, AIAA Paper2002-1274", *43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, *Denver, CO*, pp. 22-25.

Booker, A.J., Dennis, J.E., Jr, Frank, P.D., Serafini, D.B., Torczon, V. and Trosset, M.W. (1999), "A rigorous framework for optimization of expensive functions by surrogates", *Structural Optimization*, Vol. 17 No. 1, pp. 1-13.

Burgee, S., Giunta, A.A., Balabanov, V., Grossman, B., Mason, W.H., Narducci, R., Haftka, R.T. and Watson, L.T. (1996), "A coarse-grained parallel variable-complexity multidisciplinary optimization paradigm", *The International Journal of Supercomputer Applications and High Performance Computing*, Vol. 10 No. 4, pp. 269-299.

Chiba, K., Obayashi, S., Nakahashi, K. and Morino, H. (2005), "High-Fidelity multidisciplinary design optimization of wing shape for regional jet aircraft", in Coello Coello C.A., Hernández Aguirre A. and Zitzler E. (Eds*), Evolutionary Multi-Criterion Optimization, EMO 2005. Lecture Notes in Computer Science*, Vol. 3410, Springer, Berlin, Heidelberg.

Conn, A.K., Scheinberg, L.N. and Vicente, (2008), "Introduction to derivative-free optimization", *MPS-SIAM Book Series on Optimization*, SIAM, Philadelphia.

Dennis, J.E. and Torczon, V.J. (1994), "Derivative-free pattern search methods for multidisciplinary design problems", AIAA Paper 94-4349.

Economon, T.D., Palacios, F. and Alonso, J.J. (2015), "Unsteady continuous adjoint approach for aerodynamic design on dynamic meshes", *AIAA Journal*, Vol. 53 No. 9, pp. 2437-2453.

Economon, T.D., Palacios, F., Copeland, S.R., Lukaczyk, T.W. and Alonso, J.J. (2016), "SU2: an open-source suite for multiphysics simulation and design", *AIAA Journal*, Vol. 54 No. 3, pp. 828-846.

Eiben, A.E. and Smith, J. (2015), "From evolutionary computation to the evolution of things", *Nature*, Vol. 521 No. 7553, p. 476.

Emmerich, M., Deutz, A. and Klinkenberg, J.W. (2011), "Hypervolume-based expected improvement: Monotonicity properties and exact computation", *Evolutionary Computation (CEC), 2011 IEEE Congress on, IEEE.*

Epstein, B. and Peigin, S. (2004), "Robust hybrid approach to multiobjective constrained optimization in aerodynamics", *AIAA Journal*, Vol. 42 No. 8, pp. 1572-1581.

Fernandez-Godino, M.G. Park, C. Kim, N.-H. and Haftka, R.T. (2017), "Review of multi-fidelity models, under review", *Structural and Multidisciplinary Optimization*, available at: https://arxiv.org/abs/1609.07196

Fonseca, C. (1995), "Multiobjective genetic algorithms with applications to control engineering problems", PhD thesis, Department of Automatic Control and Systems Engineering, University of Sheeld, Sheeld.

Forrester, A.I.J. and Keane, A.J. (2009), "Recent advances in surrogate-based optimization", *Progress in Aerospace Sciences*, Vol. 45 Nos 1/3, pp. 50-79.

Haimes, R. and Dannenhoffer, J. (2013), "The engineering sketch pad: a solid-modeling, feature-based, web-enabled system for building parametric geometry", *21st AIAA Computational Fluid Dynamics Conference, Fluid Dynamics and Co-located Conferences*, (AIAA 2013-3073).

Han, Z.-H. and Görtz, S. (2012), "Hierarchical kriging model for Variable-Fidelity surrogate modeling", *AIAA Journal*, Vol. 50 No. 9, pp. 1885-1896.

Han, Z.H., Gortz, S. and Zimmermann, R. (2013), "Improving variable-fidelity surrogate modeling via gradient-enhanced kriging and a generalized hybrid bridge function", *Aerospace Science and Technology*, Vol. 25 No. 1, pp. 177 -189.

Hu, W.W., Saleh, K.H. and Azarm, S.S. (2012), "Approximation assisted multiobjective optimization with combined global and local metamodeling, ASME", *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Volume 3: 38th Design Automation Conference, Parts A and B*, pp. 753-764.

Hu, W., Li, M., Azarm, S. and Almansoori, A. (2011), "Multi-objective robust optimization under interval uncertainty using online approximation and constraint cuts", *Journal of Mechanical Design*, Vol. 133 No. 6, pp. 061002-1 to -061002-9.

Huang, W. (2014), "Design exploration of three-dimensional transverse jet in a supersonic crossflow based on data mining and multi-objective design optimization approaches", *International Journal of Hydrogen Energy*, Vol. 39 No. 8, pp. 3914-3925. available at: https://doi.org/10.1016/j.ijhydene.2013.12.129

Huang, W.,Z., Wang, D.B., Ingham, L., Ma, M. and Pourkashanian, (2013), "Design exploration for a single expansion ramp nozzle (SERN) using data mining", *Acta Astronautica*, Vol. 83, pp. 10-17, available at: https://doi.org/10.1016/j.actaastro.2012.09.016

Hwang, C.L. and Md Masud, A.S. (1979), "Multiple objective decision making, methods and applications: a state-of-the-art survey", *Lecture Notes in Economics and Mathematical Systems*, Springer-Verlag, Berlin Heidelberg.

Jameson, A., Schmidt, W. and 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*, 23-25 June.

Keane, A.J. (2012), "Cokriging for robust design optimization", *AIAA Journal*, Vol. 50 No. 11, pp. 2351-2364.

Kinsey, D.W. and Barth, T.J. (1984), "Description of a hyperbolic grid generation procedure for arbitrary Two-Dimensional bodies", AFWAL TM 84-191-FIMM.

Knowles, J. (2006), "ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems", *IEEE Transactions on Evolutionary Computation*, Vol. 10 No. 1, pp. 50-66.

Koziel, S. and Bekasiewicz, A. (2016), "Rapid multi-objective antenna design using point-by-point pareto set identification", *IEEE Transactions of Antennas and Propagation*, Vol. 64 No. 6.

Koziel, S. and Bekasiewicz, A. (2018), "Multi-objective design optimization of antenna structures using sequential domain patching with automated patch size determination", *Engineering Optimization*, Vol. 50 No. 2, pp. 218-234, doi: 10.1080/0305215X.2017.1311879.

Koziel, S. and Leifsson, L. (2013), "Surrogate-based aerodynamic shape optimization by variable-resolution models", *AIAA Journal*, Vol. 51 No. 1, pp. 94-106.

Koziel, S., Echeverría-Ciaurri, D. and Leifsson, L. (2011), "Surrogate-based methods", in Koziel, S. and Yang, X.S. (Eds), *Computational Optimization, Methods and Algorithms, Series: Studies in Computational Intelligence*, Springer-Verlag, Berlin Heidelberg, pp. 33-60.

Laurenceau, J. and Sagaut, P. (2008), "Building efficient response surfaces of aerodynamic functions with kriging and cokriging", *AIAA Journal*, Vol. 46 No. 2, pp. 498-507.

Leifsson, L., Koziel, S. and Tesfahunegn, Y.A. (2016), "Multiobjective aerodynamic optimization by variable-fidelity models and response surface surrogates", *AIAA Journal*, Vol. 54 No. 2, pp. 531-541.

Li, M., Li, G. and Azarm, S. (2008), "A kriging metamodel assisted multi-objective genetic algorithm for design optimization", *Journal of Mechanical Design*, Vol. 130 No. 3, pp. 031401-1 to -031401-10.

Liem, R.P., Martins, Jrra. and Kenway, G.K. (2017), "Expected drag minimization for aerodynamic design optimization based on aircraft operational data", *Aerospace Science and Technology*, Vol. 63, pp. 344-362.

Mengistu, T. and Ghaly, W. (2008), "Aerodynamic optimization of turbomachinery blades using evolutionary methods and ANN-based surrogate models", *Optimization and Engineering*, Vol. 9 No. 3, pp. 239-255.

Nemec, M., Zingg, D.W. and Pulliam, T.H. (2004), "Multipoint and multi-objective aerodynamic shape optimization", *AIAA Journal*, Vol. 42 No. 6, pp. 1057-1065.

Ou, M., Yan, L., Huang, W. and Zhang, T.T. (2019), "Design exploration of combinational spike and opposing jet concept in hypersonic flows based on CFD calculation and surrogate model", *Acta Astronautica*, Vol. 155, pp. 287-301, available at: https://doi.org/10.1016/j.actaastro.2018.12.012

Peherstorfer, B. Willcox, K. and Gunzburger, M. (2017), "Survey of multifidelity methods in uncertainty propagation, inference, and optimization", SIAM Review, accepted.

Poli, R., Kennedy, J. and Blackwell, T. (2007), "Particle swarm optimization", *Swarm Intelligence*, Vol. 1 No. 1, pp. 1935-3820.

Qian, P.Z. and Wu, C.J. (2008), "Bayesian hierarchical modeling for integrating low-accuracy and high-accuracy experiments", *Technometrics*, Vol. 50 No. 2, pp. 192-204.

Queipo, N.V., Haftka, R.T., Shyy, W., Goel, T., Vaidyanathan, R. and Tucker, P.K. (2005), "Surrogate-based analysis and optimization", *Progress in Aerospace Sciences*, Vol. 41 No. 1, pp. 1-28.

Rao, S.S. (1996), *Engineering Optimization: Theory and Practice*, 3rd ed., John Wiley and Sons, New York, NY.

Rathinam, M. and Petzold, L. (2003), "A new look at proper orthogonal decomposition", *SIAM Journal on Numerical Analysis*, Vol. 41 No. 5, pp. 1893-1925.

Ren, Z., Thelen, A.S., Amrit, A., Du, X., Leifsson, L., Tesfahunegn, Y.A. and Koziel, S. (2016), "Application of multifidelity optimization techniques to benchmark aerodynamic design problems", *54th AIAA Aerospace Sciences Meeting, San Diego, CA*.

Rozza, G., Huynh, D.B.P. and Patera, A.T. (2008), "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, pp. 229-275.

Scholkopf, B. and Smola, A.J. (2001), *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, Cambridge.

Shan, S. and Wang, G. (2005), "An efficient pareto set identification approach for multiobjective optimization on Black-Box functions", *Journal of Mechanical Design*, Vol. 127 No. 5, pp. 866-874.

Shimoyama, K., Sato, K., Jeong, S. and Obayashi, S. (2012), "Comparison of the criteria for updating kriging response surface models in multi-objective optimization", *Evolutionary Computation (CEC), 2012 IEEE Congress on.*

Spalart, P., R, Allmaras and S, R. (1992), "A one equation turbulence model for aerodynamic flows, AIAA-Paper-92-0439", *38th AIAA Aerospace Sciences Meeting and Exhibit, Reno*, 6-9 January.

Storn, R. and Price, K. (1997), "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces",*Journal of Global Optimization*, Vol. 11 No. 4, pp. 341-359.

Teckentrup, A.L., Jantsch, P., Webster, C.G. and Gunzburger, M. (2015), "A multilevel stochastic collocation method for partial differential equations with random input data", *SIAM/ASA Journal on Uncertainty Quantification*, Vol. 3 No. 1, pp. 1046-1074.

Wang, Z., Huang, W. and Li, Y. (2014), "Multidisciplinary design optimization approach and its application to aerospace engineering", *Chinese Science Bulletin*, Vol. 59 No. 36, pp. 5338, doi: 10.1007/s11434-014-0671-1.

Xu, G., Liang, X., Yao, S., Chen, D. and Li, Z. (2017), "Multi-objective aerodynamic optimization of the streamlined shape of High-Speed trains based on the kriging model", *PLoS One*, Vol. 12 No. 1, p. e0170803, available at: https://doi.org/10.1371/journal.pone.0170803

Yang, X.S. (2009), "Firefly algorithms for multimodal optimization, in: Stochastic algorithms: foundations and applications, Saga 2009", *Lecture Notes in Computer Sciences*, Vol. 5792, pp. 169-178.

Yang, X.S. and Deb, S. (2014), "Cuckoo search: recent advances and applications", *Neural Computing and Applications*, Vol. 24 No. 1, pp. 169-174.

Zhang, T.T., Wang, Z.G., Huang, W. and Li, Y. (2016a), "Parameterization and optimization of hypersonic-gliding vehicle configurations during conceptual design", *Aerospace Science and Technology*, Vol. 58, pp. 225-234, available at: https://doi.org/10.1016/j.ast.2016.08.020

Zhang, T.T., Huang, W., Wang, Z.G. and Li, Y. (2016b), "A study of airfoil parameterization, modeling, and optimization based on the computational fluid dynamics method", *Journal of Zhejiang University – Science A*, Vol. 17 No. 8, pp. 632-645, doi: 10.1631/jzus.A1500308s.

Further reading

March, A. and Willcox, K. (2012), "Constrained multifidelity optimization using model calibration", *Structural and Multidisciplinary Optimization*, Vol. 46 No. 1, pp. 93-109.

Corresponding author

Leifur Leifsson can be contacted at: leifur@iastate.edu