

Gradient-based Pareto Front Approximation Applied to Turbomachinery Shape Optimization

ILIAS VASILOPOULOS*[†]
iliias.vasilopoulos@rolls-royce.com

VARVARA G. ASOUTI[†]
vasouti@mail.ntua.gr

KYRIAKOS C. GIANNAKOGLOU[†]
kgianna@central.ntua.gr

MARCUS MEYER*
marcus.meyer@rolls-royce.com

*Rolls-Royce Deutschland, Blankenfelde-Mahlow, Germany

[†]National Technical University of Athens (NTUA), School of Mechanical Engineering, Parallel CFD & Optimization Unit, Athens, Greece

Abstract

Multi-objective optimization has been rising in popularity, especially within an industrial environment, where several cost functions often need to be considered during the design phase. Traditional gradient-free approaches, such as evolutionary algorithms, can be employed to compute a front of equally suitable solutions a designer can choose from, with a high computational cost though, particularly in high-dimensional design spaces. In this paper, a gradient-based algorithm is developed for efficiently tracing the Pareto front in bi-objective aerodynamic shape optimization problems, where an adjoint method is used for the computation of the objective functions' gradients with respect to (w.r.t.) the design variables. After obtaining a starting point on the front, a prediction-correction approach is employed to compute new Pareto points. Satisfying the Karush-Kuhn-Tucker (KKT) conditions provides a prediction for the next point, which is, then, corrected by solving a minimum distance problem. The prediction step, though, requires the costly computation of the Hessian matrix. This is avoided here by using the BFGS [1] technique. The proposed method is first demonstrated in a less expensive lift/drag optimization of an isolated airfoil and, then, applied to the bi-objective optimization of a 3D compressor stator. The extension of the proposed method to cases with more than two objectives is straightforward, on condition that an algorithm is found to coordinate the way the Pareto front is swept.

I. INTRODUCTION

Optimization in aerodynamics and engineering, in general, may involve several contradictory objectives. These multi-objective optimization (MOO) problems can be solved using stochastic approaches, such as Evolutionary Algorithms (EAs), which are able to capture the front of non-dominated solutions with a single run. EAs treat the objective function evaluation tool as a "black box" and, thus, no additional software development is required. Provided that they are run for a sufficiently long time, EAs are able to find the global optima, by computing convex, non-convex and discontinuous Pareto fronts [2, 3]. However, the main disadvantage associated with EAs is that their computational cost scales with the number of design variables [4]. As a remedy to this problem, metamodel-assisted EAs (i.e. EAs regularly implementing low-cost surrogate evaluation models within the evolution) and/or dimensionality reduction techniques using, for instance, Principal Component Analysis [5] can be applied.

On the other hand, gradient-based approaches deal with MOO problems by performing a series of single-objective optimization (SOO) runs. The most popular gradient-based strategy is the weighted-sum approach, in which a convex combination of the individual objectives is minimized. Differently weighting the objectives leads to different non-dominated solutions and, thus, the process is repeated until the Pareto front is adequately populated. However, a uniform spread of weights rarely results in a good distribution of points along the Pareto front. If the Pareto front is non-convex, none of the combinations of weights may compute internal front points.

In order to approximate the Pareto front with a number of well-distributed optimal points, several approaches have been proposed. The Normal Boundary Intersection (NBI) method, where the Pareto front is sought by projecting points lying on the convex hull of local minima of each individual objective, is presented in [6]. A uniformly-spaced grid of points on the convex hull results in a good distribution of Pareto optimal solutions. Although this method is applicable to high objective space dimensions, it is not guaranteed that the whole Pareto front is found whereas special treatment is required at the non-convex parts of the front.

These issues of the NBI can be addressed by the adaptive weighted-sum method [7], where, instead of prescribing a-priori weights to populate the Pareto front, weights are dynamically modified. A small number of SOO problems are initially solved for given combinations of weights and, then, the unexplored Pareto regions are populated by specifying additional inequality constraints. The extra computational cost as well as the dependency on some user-defined parameters are the main weaknesses of this method. In addition, extension to problems with more than two objectives is not straightforward. An improvement to the NBI method for high objective space dimensions is proposed in [8], successively populating the inner parts of bounded Pareto fronts. For such cases, the boundary of the Pareto front can be computed using objective reduction techniques [9].

A different scalarization technique using steepest descent is described in [10], where convex and non-convex well-distributed Pareto fronts are generated. This is also achieved in [11] by augmenting the KKT conditions with additional objective-space-based distance constraints and, then, using Newton's method to solve the coupled system of equations, which drive the set of points towards Pareto optimality and uniform spacing simultaneously. The required (exact) Hessian matrix is derived by a hyper-dual numbers implementation, at very high cost, though, for practical applications.

Target or reference point methods have been used in optimal flow control [12] as a robust way of moving from one Pareto point to another. They require the successive solution of many SOO problems, which is computationally intensive and, as a remedy to this, reduced-order approaches based on Proper Orthogonal Decomposition have been proposed [13].

An alternative to scalarization techniques are the curve continuation strategies which aim at moving from one Pareto point to a neighbouring one by taking into account the local front curvature. Such an approach is introduced in [14] in the form of a prediction-correction scheme. A constrained minimization problem is defined, where equality constraints are imposed to all but one objective function which is to be minimized. Formulating the KKT conditions and satisfying the requirements of the implicit function theorem, a linear system is derived, the solution of which can be used to predict a new Pareto point. Due to the non-linearity of the problem, an iterative correction step is required to bring the point onto the Pareto front. The corrector can be any constrained optimization solver, such as sequential quadratic programming (SQP). Although this approach has the advantage of controlling the distance between two Pareto points, this depends on the user-defined value of the constraints, a bad choice of which can lead to the SQP problem being unsolvable. Moreover, the need for the Hessian matrix can be a limiting factor regarding performance. In [15], a number of alternatives to computing the Hessian for such a curve

continuation approach are presented. These include Hessian approximations using BFGS and the computation of Hessian-vector products, instead of the Hessian itself, based on the truncated Newton method [16].

In this work, a different gradient-based prediction-correction algorithm is developed for efficiently moving on the Pareto front. The prediction phase is inspired by the corresponding step used in [14], while key differences exist in the correction phase. Instead of solving a constrained SOO problem at each correction step, an unconstrained formulation is proposed, where the Pareto front is traced by minimizing the distance (in the objective space) from properly defined target points. The distance function is treated as the objective in the formulation of the KKT stationarity condition, computing the minimum distance solution. The required gradients are obtained using either discrete or continuous adjoint; in the results, Sect. III, the first case is handled using continuous [17] whereas the second one uses the discrete [18] adjoint. BFGS is selected as the optimization method in the correction step, providing thus a Hessian approximation to be utilized in the subsequent prediction step. After demonstrating the proposed method on an isolated airfoil, its industrial applicability is indicated using a 3D turbomachinery component which is presented in detail.

II. METHOD FORMULATION

A MOO problem is defined as

$$\begin{aligned} \min_{\mathbf{b} \in B} F(\mathbf{b}) &:= [f_1(\mathbf{b}) \dots f_M(\mathbf{b})]^T \\ \text{s.t.} \quad \mathbf{c}(\mathbf{b}) &= 0 \\ \mathbf{g}(\mathbf{b}) &\geq 0 \end{aligned} \quad (1)$$

where $F : \mathbb{R}^N \rightarrow \mathbb{R}^M$, $\mathbf{c} : \mathbb{R}^N \rightarrow \mathbb{R}^{M_1}$, $\mathbf{g} : \mathbb{R}^N \rightarrow \mathbb{R}^{M_2}$ are the objective and constraint functions, $B := \{\mathbf{b} \in \mathbb{R}^N \mid \mathbf{c}(\mathbf{b}) = 0, \mathbf{g}(\mathbf{b}) \geq 0\}$ denotes the feasible solutions' set, N is the number of design variables and $M > 1$ the number of objectives. Since a single $\mathbf{b}^* \in B$ cannot minimize all objective functions f_i at the same time, the optimality definition is expanded in the Pareto sense. For $\mathbf{b}_1, \mathbf{b}_2 \in B$, $F(\mathbf{b}_1)$ is said to dominate $F(\mathbf{b}_2)$ if:

$$\forall i \in \{1, \dots, M\} : f_i(\mathbf{b}_1) \leq f_i(\mathbf{b}_2) \quad \text{and} \quad \exists j \in \{1, \dots, M\} : f_j(\mathbf{b}_1) < f_j(\mathbf{b}_2) \quad (2)$$

A solution \mathbf{b}^* is called Pareto optimal if it is non-dominated [6]. Then, the Pareto set P_s and Pareto front P_f are defined as $P_s = \{\mathbf{b}^* \in B \mid \mathbf{b}^* \text{ is Pareto optimal}\}$ and $P_f = F(P_s)$, respectively.

As discussed in Sect. I, various methods exist to solve problem (1). In this work, a target point approach is enhanced using a prediction technique. After obtaining an initial point on the Pareto front (upon completion of the so-called "Go-to-Pareto" step), an infeasible target point is automatically computed in the objective space and the Euclidean distance of the objective vector to that target vector is minimized, resulting in a new Pareto point. This process is repeated until the Pareto front is adequately populated. Moving between neighbouring Pareto points is accelerated using a prediction-correction technique, in which the prediction is driven by approximate Hessian information.

i. The "Go-to-Pareto" Step

To initialize the method, a first point \mathbf{b}_0^* that belongs to the Pareto front must be found. This is achieved by minimizing a weighted-sum of the M objectives

$$\min_{\mathbf{b} \in B} F_w(\mathbf{b}) := \mathbf{w} \cdot \mathbf{F}(\mathbf{b}) = \sum_{j=1}^M w_j f_j(\mathbf{b}) \quad (3)$$

subject to the constraints of problem (1), where $w_j \geq 0 \forall j \in \{1, \dots, M\}$. Populating the Pareto front by performing a series of "Go-to-Pareto" steps (3), using different weight (\mathbf{w}) combinations, is computationally expensive and non-applicable to non-convex problems, thus, an alternative approach is proposed herein.

ii. The "Move-on-Pareto" Steps

Moving along the Pareto front can be achieved by means of appropriately selected target points. This technique can be found in the literature with a wide variety of slightly different formulations [12, 19]. At each step, a target point $\mathbf{T}_i = [\hat{f}_1 \dots \hat{f}_k \dots \hat{f}_M]^T \in \mathbb{R}^M$ which does not belong to the feasible solutions' set has to be defined in the objective space. Here, this is done by shifting an already computed Pareto point $\mathbf{F}(\mathbf{b}_{i-1}^*)$ towards a direction that reduces one objective while retaining the values of the other objectives, i.e. by defining the target point as

$$\mathbf{T}_i := [f_1(\mathbf{b}_{i-1}^*) \dots f_k(\mathbf{b}_{i-1}^*) + \Delta \hat{f}_k \dots f_M(\mathbf{b}_{i-1}^*)]^T \quad (4)$$

where $\Delta \hat{f}_k < 0$. In what follows, points marked with (*) belong to the sought Pareto front. \mathbf{T}_i is an infeasible point, since it would otherwise dominate $\mathbf{F}(\mathbf{b}_{i-1}^*)$ according to the Pareto optimality condition (2). Then, a new (i^{th}) Pareto point is obtained by solving the distance minimization problem (with non-dimensional values for the objective components f_j):

$$\min_{\mathbf{b} \in B} \frac{1}{2} \|\mathbf{F}(\mathbf{b}) - \mathbf{T}_i\|_2^2 := \frac{1}{2} \sum_{j=1}^M (f_j(\mathbf{b}) - \hat{f}_j)^2 \quad (5)$$

subject to the constraints of problem (1). Thus, the solution to problem (5) is geometrically defined as the projection of \mathbf{T}_i onto the Pareto front. By updating the target position based on Pareto points already found, additional points on the Pareto front can be computed one after the other. In general, this "Move-on-Pareto" technique is not restricted to low objective space dimensions, and it amounts to finding an algorithm for redefining the targets in such a way that the Pareto front is adequately populated.

The simplicity and robustness of the proposed approach are the key ingredients which support its implementation within an industrial workflow. However, the method's efficiency cannot be neglected, since solving problem (5) multiple times is cost intensive. In this paper, this is tackled by adding a prediction step (for the next Pareto point), which provides a good initial guess and accelerates the convergence significantly.

The prediction step is inspired by the Pareto curve continuation method presented in [14]. The main difference lies in the fact that the implicit function theorem is applied to problem (5) instead of (3). Further consideration of constraints is omitted, since only unconstrained optimization problems are addressed in this work. However, there is no gap in the method, since the generic formulation (including constraints) can be derived following [14] and solved using, for instance, the SQP method [15].

Let \mathbf{b}_i^* be the solution to problem (5) for the target point T_i as defined in Eq. (4). This should satisfy the KKT conditions:

$$\nabla_{\mathbf{b}} \left(\frac{1}{2} \|\mathbf{F}(\mathbf{b}) - T_i\|_2^2 \right) = 0 \Leftrightarrow \underbrace{\sum_{j=1}^M (f_j(\mathbf{b}) - \hat{f}_j) \nabla_{\mathbf{b}} f_j(\mathbf{b})}_{G(\mathbf{b}, \hat{f}_k)} = 0 \quad (6)$$

In order for a point to remain on the Pareto front, the total derivative of $G(\mathbf{b}, \hat{f}_k)$ w.r.t. the target (i.e. \hat{f}_k) should be zero:

$$\frac{\delta G(\mathbf{b}, \hat{f}_k)}{\delta \hat{f}_k} = 0 \Leftrightarrow \frac{\partial G(\mathbf{b}, \hat{f}_k)}{\partial \hat{f}_k} + \frac{\partial G(\mathbf{b}, \hat{f}_k)}{\partial \mathbf{b}} \frac{\delta \mathbf{b}}{\delta \hat{f}_k} = 0$$

or

$$\left[\sum_{j=1}^M \left(\nabla_{\mathbf{b}} f_j (\nabla_{\mathbf{b}} f_j)^T + (f_j - \hat{f}_j) \nabla_{\mathbf{b}}^2 f_j \right) \right] \frac{\delta \mathbf{b}}{\delta \hat{f}_k} = \nabla_{\mathbf{b}} f_k \quad (7)$$

After solving the linear system of Eq. (7) for $\frac{\delta \mathbf{b}}{\delta \hat{f}_k}$ at the current point \mathbf{b}_i^* , a prediction \mathbf{b}_{i+1}^{pred} of the next point \mathbf{b}_{i+1}^* on the Pareto front is given by a first-order Taylor expansion

$$\mathbf{b}_{i+1}^{pred} = \mathbf{b}_i^* + \left. \frac{\delta \mathbf{b}}{\delta \hat{f}_k} \right|_i \Delta \hat{f}_k$$

It should be noted that, in Eq. (7), the Hessian matrices $\nabla_{\mathbf{b}}^2 f_j$ are required and, thus, they should be computed or approximated.

The new point \mathbf{b}_{i+1}^* is obtained by solving the distance minimization problem (5) from an updated target point T_{i+1} while using \mathbf{b}_{i+1}^{pred} as an initial guess. This step can be considered as the correction to the previous prediction. Mathematically speaking, a T_{i+1} consistent with the above prediction is obtained by displacing T_i using the perturbation vector $[0 \dots \Delta \hat{f}_k \dots 0]^T$. However, for small values of $\Delta \hat{f}_k$, displacing any point that lies along the line defined by T_i and $\mathbf{F}(\mathbf{b}_i^*)$ towards that direction gives a different target point, which, when used in problem (5), should result in a similar Pareto solution. In practice, this was verified during the preliminary testing of the method and, thus, defining every new target point T_{i+1} w.r.t. the previously computed Pareto point $\mathbf{F}(\mathbf{b}_i^*)$, as in Eq. (4), was found to be the best option, especially when combined with a constant $\Delta \hat{f}_k$.

In [12], an alternative way of defining target points is proposed using a tangent approximation, to deal with Pareto fronts of highly varying steepness.

iii. Bi-objective Optimization Algorithm

In this work, the proposed method is applied only to problems with two objectives ($M = 2$), as illustrated in Fig. 1, using Algorithm 1. A BFGS-based method was developed in order to solve the SOO problems (3) (i.e. to obtain an initial Pareto point) and (5), and at the same time provide an approximation to the Hessian matrices needed in Eq. (7). The BFGS implementation includes a line search algorithm which aims at accelerating the optimization by satisfying at each step the Armijo condition [1]. This condition assures that the chosen step length sufficiently decreases the function to be minimized. Thus, finding an appropriate step length comes at a cost of up to two additional function evaluations, while reducing the required number of the (more expensive) BFGS steps. The gradients needed to drive the optimization method are obtained by adjoint solvers.

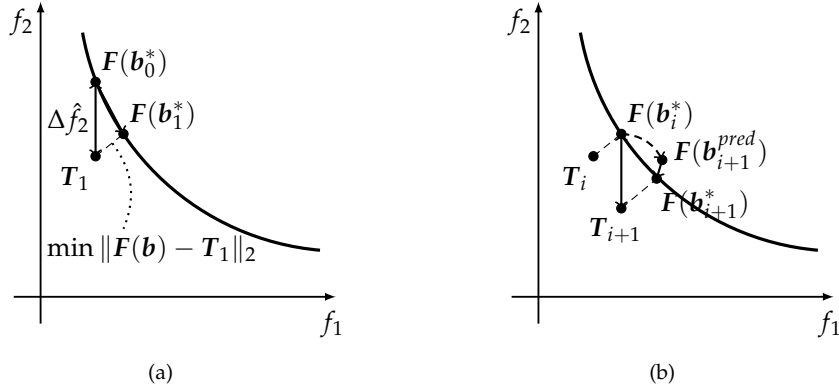


Figure 1: Proposed method in the bi-objective space: (a) Computation of Pareto point \mathbf{b}_1^* by solving the distance minimization problem (5) starting from \mathbf{b}_0^* (without prediction); (b) Definition of new target point T_{i+1} and acceleration of the method using a prediction step after solving the linear system of Eq. (7)

Algorithm 1 (for bi-objective problems)

Read: Reference point \mathbf{b}_0 , weights $[w_1 \ w_2] = [1 \ 0]$, number of steps n_s (along the Pareto front), step length $\Delta \hat{f}_2 < 0$

- 1: Solve optimization problem (3) $\rightarrow \mathbf{b}_0^*$ (initial Pareto point: $f_1(\mathbf{b}_0^*) = f_1^{min}$)
 - 2: Define the 1st target point (4): $T_1 = F(\mathbf{b}_0^*) + [0 \ \Delta \hat{f}_2]^T$
 - 3: Starting from \mathbf{b}_0^* , solve optimization problem (5) $\rightarrow \mathbf{b}_1^*$ (using BFGS)
 - 4: Store $\nabla_b f_1, \nabla_b f_2$ (computed by adjoint) and $\nabla_b^2 f_1, \nabla_b^2 f_2$ (approximated by BFGS) at \mathbf{b}_1^*
 - 5: $i = 1$
 - 6: **while** ($i < n_s$ **and** $\|F(\mathbf{b}_i^*) - F(\mathbf{b}_{i-1}^*)\|_2 > \varepsilon$) **do**
 - 7: Solve linear system (7) $\rightarrow \left. \frac{\delta \mathbf{b}}{\delta \hat{f}_2} \right|_i$
 - 8: Define new target point $T_{i+1} = F(\mathbf{b}_i^*) + [0 \ \Delta \hat{f}_2]^T$
 - 9: $\mathbf{b}_{i+1}^{pred} = \mathbf{b}_i^* + \left. \frac{\delta \mathbf{b}}{\delta \hat{f}_2} \right|_i \Delta \hat{f}_2$ (prediction step)
 - 10: Starting from \mathbf{b}_{i+1}^{pred} , solve (5) $\rightarrow \mathbf{b}_{i+1}^*$ (correction)
 - 11: Store $\nabla_b f_1, \nabla_b f_2$ and $\nabla_b^2 f_1, \nabla_b^2 f_2$ at \mathbf{b}_{i+1}^*
 - 12: $i = i + 1$
 - 13: **end**
-

In order to quantify the method's efficiency improvement due to the addition of prediction steps, in Sect. III, the first "Move-on-Pareto" step is always performed using exclusively the iterative correction step, while omitting the prediction step used in any subsequent Pareto point search, as shown in Fig. 1(a). A comparison is made between the cost of computing \mathbf{b}_1^* and the rest of the Pareto points.

The second condition in line 6 of Algorithm 1 is employed to ensure that, in case the chosen number of steps n_s is too big, the algorithm will terminate when the distance between two subsequent Pareto points becomes smaller than ε . Herein, $\varepsilon = 0.1|\Delta \hat{f}_2|$ is selected.

III. APPLICATIONS

i. Shape Optimization of an Isolated Airfoil

Before moving on to the 3D turbomachinery application, the proposed method is demonstrated in the lift/drag optimization of the isolated NACA0012 airfoil. The initial geometry along with the applied parameterization using two Bezier curves [20] are plotted in Fig. 2.

The symmetric baseline shape is parameterized with 14 Control Points (CPs), 6 out of which remain fixed and the rest are allowed to move in the y direction, resulting in 8 "free" design variables. In the absence of any other constraint (such as a volume constraint), reasonable bounds are selected for these parameters in order to avoid non-realistic airfoil shapes during the optimization. Then, an unstructured C-type mesh consisting of $\sim 75k$ hybrid elements is generated. At each design iteration, the mesh is updated to match the varying airfoil geometry using an RBF mesh morpher [21].

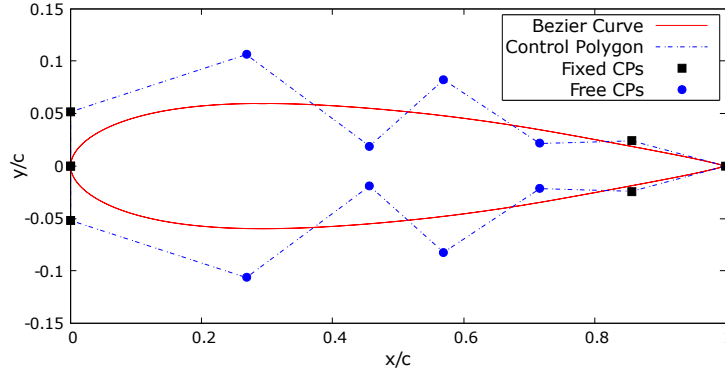


Figure 2: Parameterization of baseline airfoil geometry using Bezier curves. The "fixed" CPs are not displaced during the optimization, whereas the "free" ones can move vertically (one design variable per "free" CP)

Regarding the flow conditions, the freestream flow angle, Mach and Reynolds numbers are: $\alpha_\infty = 3^\circ$, $M_\infty = 0.15$ and $Re_\infty = 3 \times 10^6$, respectively. Flow simulations are performed using a CFD code for unstructured grids [22], running on graphics processing units (GPUs). The steady-state Reynolds-averaged Navier-Stokes (RANS) equations for compressible flow are solved using a vertex-centered finite-volume discretization scheme; turbulence is resolved with the one-equation Spalart-Allmaras (SA) model [23]. Lift and drag gradients w.r.t. the design variables are computed using the continuous adjoint method [17].

The MOO problem is defined as

$$\min \begin{cases} f_1(\mathbf{b}) = -c_L(\mathbf{b}) \\ f_2(\mathbf{b}) = c_D(\mathbf{b}) \end{cases}$$

where c_L and c_D the lift and drag coefficients of the airfoil. Applying Algorithm 1 with $n_s = 10$ and $\Delta \hat{f}_2 = -0.1$, the results plotted in Fig. 3 are obtained. Here, the aim is not to generate the whole Pareto front (not both corner points were sought), but rather to verify the applicability of the adjoint-based approach. Additionally, in order to perform the MOO within an acceptable computational cost, the convergence criteria of the SOO problems solved in the correction steps are reasonably relaxed. The cost is measured in terms of the equivalent flow solutions (EFS; one EFS stands for the cost of solving the flow equations using the CFD solver of the case)

required for the primal and adjoint simulations, so the "Go-to-Pareto" step involved 30 EFS, the first "Move-on-Pareto" step (without prediction) needed 18 EFS and each subsequent step (using prediction-correction) required on average ~ 12 EFS, resulting in a total computational cost of 159 EFS for obtaining 11 Pareto points. The cost breakdown demonstrates the benefit of the proposed method in terms of efficiency.

In order to verify the results, a metamodel-assisted EA-based (MAEA) optimization software (EASY [24], developed by the NTUA) was employed to generate the Pareto front. As shown in Fig. 3, the adjoint-based solution set is in good agreement with the MAEA one (only part of which is shown here), both of which are sufficiently approximating the front. More Pareto points are computed by the MAEA optimization at a slightly higher total cost (175 EFS) though. Due to the relaxed termination criteria, the set of non-dominated solutions obtained by the adjoint method is an approximation to the (exact) Pareto front; this is however computed faster than the MAEA front, even for a rather small value of design variables ($N = 8$).

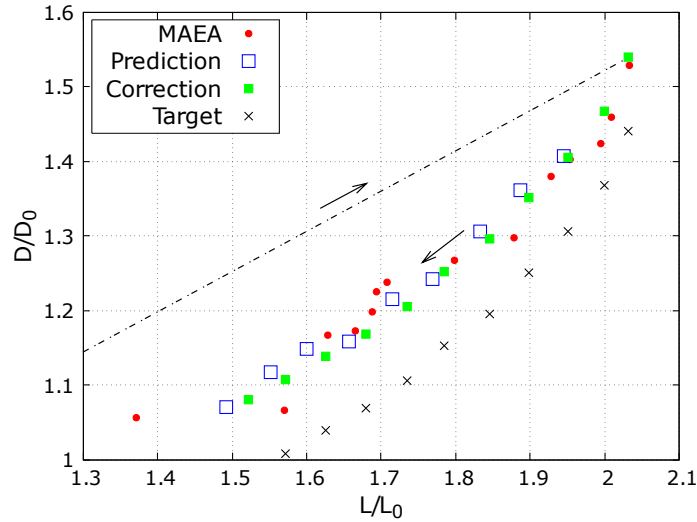


Figure 3: Solution of the lift/drag problem for an isolated airfoil: Pareto front approximation by 11 optimal points obtained using the proposed method and comparison with MAEA results. The dashed line stands for the "Go-to-Pareto" step

ii. Shape Optimization of a Compressor Stator

Herein, the proposed method is applied to the bi-objective optimization of a 3D compressor stator.

ii.1 Description and CFD setup

This subsection presents the case setup with focus on the steps that have to be considered in order to perform the CFD simulation. A similar setup has been previously used in [25].

The initial stator geometry is available online [26]. The goal is to optimize this baseline stator (BS) w.r.t. two criteria, which leads to a MOO problem. The first objective to be minimized is the total pressure losses between the inlet and exit, defined by the loss coefficient:

$$\omega = \frac{p_{t,I} - p_{t,E}}{p_{t,I} - p_I}$$

where p and p_t denote the mass-averaged static and total pressure computed at the inlet (I) and exit (E) of the CFD domain. The second objective to be minimized is the flow angle deviation from the axial direction at the stator exit:

$$\alpha_E = \sqrt{\frac{\int_{A_E} \dot{m} \alpha^2 dA_E}{\int_{A_E} \dot{m} dA_E}}, \quad \alpha = \cos^{-1} \left(\frac{u}{\sqrt{u^2 + v^2 + w^2}} \right)$$

where u , v , w denote the Cartesian velocity components and \dot{m} the mass flow rate. Only the design point operating conditions are considered, with a uniform inlet whirl angle profile of 42° . These two objectives are contradicting since the higher the turning – which is required here to achieve a more axial outflow – the higher the losses. Moreover, several geometric constraints (such as minimum leading edge (LE) and trailing edge (TE) radii, constant axial chord length, etc.) are taken into account during the optimization process. Their detailed definition, as well as information about the boundary conditions of the case, can be found in [26].

The BS geometry is parameterized using the Rolls-Royce in-house blade design tool [27] and is constructed by 21 individual 2D profile sections radially stacked from hub to tip. A curve in the radial direction is defined by axial and circumferential shifts, and the sections are stacked accordingly, resulting in the final 3D blade shape [28]. For each profile contour definition, a section build-up based on the superposition of a camber-line angle $\beta(x)$ and a thickness distribution $T(x)$ is applied [29]. In the present work, the normalized distributions plotted in Fig. 4 are parameterized by B-splines [20], using 7 CPs for the non-dimensional camber-line angle distribution and 9 CPs for the normalized thickness distribution, but they are kept constant during the MOO.

From the 21 available 2D profiles, 7 are selected as design sections and the rest are obtained by spline interpolation. Thus, the 5 design parameters (r_I , r_E , β_I , β_E and T_{max}) per design section and 2 stacking parameters for each section's axial and circumferential shift, excluding hub and tip, sum to 45 design variables in total. Using such a CAD-based parameterization, the geometric constraints of the case can a-priori be guaranteed by properly bounding the design space.

Figure 5 shows the block-structured hexahedral mesh with ~ 2 million nodes that was generated using an in-house meshing tool [30], after conducting a mesh refinement study to achieve a mesh-independent computation of the objectives. The maximum y^+ value of the first series of nodes off the walls is below 4, which is in conformity with the adaptive wall function technique [31] used in the CFD simulation.

A steady-state RANS compressible flow solver with the SA turbulence model is used to compute the flow. This is part of the Rolls-Royce in-house CFD suite of codes [32], which have

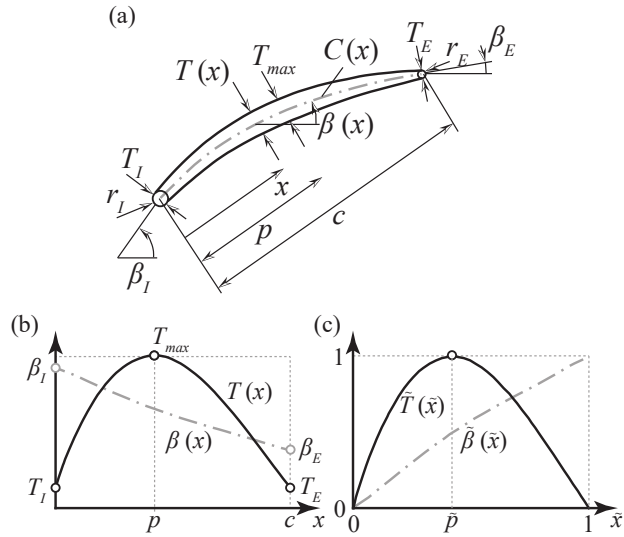


Figure 4: Parameterization of blade section: (a) Profile Skeleton; (b) Camber-line angle and thickness distributions; (c) Non-dimensional distributions

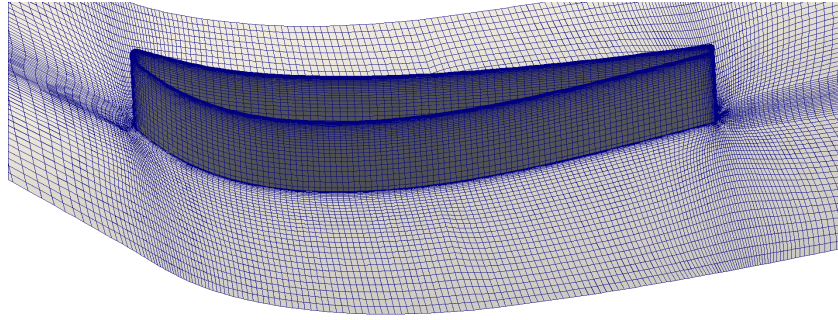


Figure 5: CFD mesh of the BS mid-span section

been extensively validated and applied to several industrial applications [33]. The non-linear flow solver uses a vertex-centered finite-volume discretization scheme and the 5-stage Runge-Kutta method [34] (for the pseudo-time-stepping) is accelerated by a block-Jacobi preconditioner and a geometric multigrid technique [35]. Figure 6(a) illustrates the close-to-surface streamlines after the flow simulation for the BS geometry has converged. It can be seen that corner flow separations occur at both hub and tip which, then, expand along the suction side of the blade. This is common for highly loaded compressor stators and it was also experimentally observed in [36], where this BS geometry was manufactured and measured. These separations contribute to a loss of $\omega_0 = 0.0764$, while the flow turning achieved by the BS blade is such that $\alpha_{E,0} = 3.74^\circ$. In [36], the presented CFD setup was experimentally verified as an accurate simulation tool for this compressor stator application.

The HYDRA discrete adjoint solver [18] is used to provide the volume sensitivities (i.e. the change in objective function w.r.t. a volume mesh node perturbation). Surface sensitivities (i.e. the change in objective function w.r.t. a surface mesh node perturbation) are computed by applying the inverse operation of a spring-based mesh deformation algorithm to the volume sensitivities [38]. Then, the surface sensitivity map is obtained by projecting each surface node's sensitivity vector onto its corresponding boundary normal. This is visualized for the BS in Fig. 6(b) where

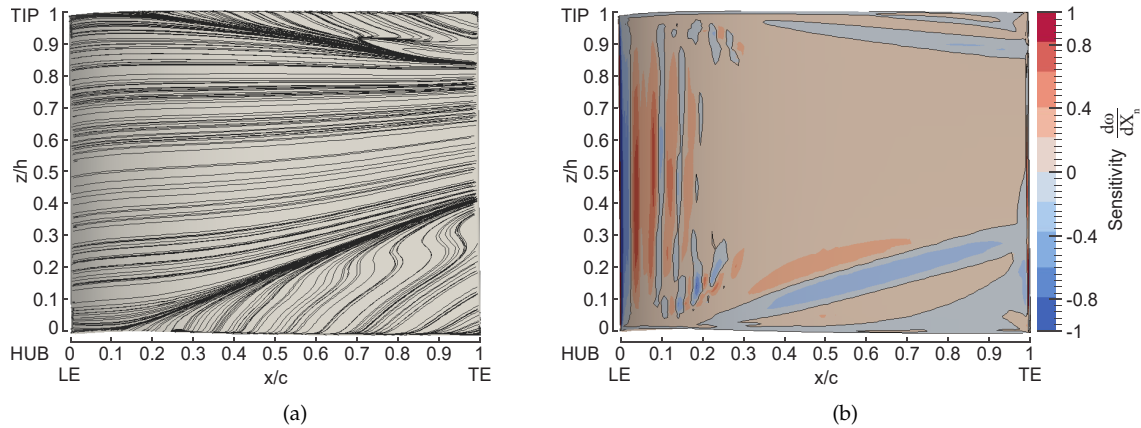


Figure 6: CFD results on the suction side of the BS blade: (a) Surface streamlines of converged flow; (b) Adjoint sensitivity map of ω w.r.t. surface normal displacements

ω is used as the objective. The sensitivity map shows which areas should be modified in order to suppress the corner flow separations and, thus, reduce the losses. Here, blue areas should be pulled outwards and red pushed inwards to achieve a decrease in ω . The black line depicts the zero sensitivity line.

Coupling the mesh sensitivities $\frac{d\omega}{dX_S}$, $\frac{d\alpha_E}{dX_S}$ with the geometric sensitivities $\frac{dX_S}{db}$ (using the chain rule) results in the adjoint gradients $\frac{d\omega}{db}$, $\frac{d\alpha_E}{db}$, where \mathbf{b} denotes the blade design parameters. The accuracy of these gradients was verified as in the previous study [37]. The required computation of geometric sensitivities ($\frac{dX_S}{db}$) for each design parameter is done using second-order FD. Although the computational cost of FD for the geometry part scales with the number of design parameters, the total cost for the 45 parameters used in this case is negligible compared to the cost of the primal and adjoint solutions.

ii.2 Computing the initial Pareto point

The first Pareto point required to initialize the MOO process is computed as described in step 1 of Algorithm 1. Hence, an initial optimal stator (OPS₀) blade is sought, which is obtained by minimizing exclusively w.r.t. the first objective ω . In order to perform this SOO for the "Go-to-Pareto" step, an automated workflow was developed within the Rolls-Royce design process for this compressor stator application. This SOO workflow is also employed during the correction steps of the "Move-on-Pareto" process (following subsection), using a different minimization function.

The BFGS optimization method has come up with a leaned (bow-shaped) blade; Fig. 8-(b), which is more loaded at mid-span and less loaded at hub and tip, thus reducing the (still existing) corner flow separations. This is also illustrated in Fig. 9, where the total pressure loss coefficient distribution at an axial cross-section just behind the TE is plotted for the BS (a) and OPS₀ (b) blades. Especially the tip corner separation is almost fully suppressed. Thickness minimization (up to the bound value), LE/TE optimization (increased radii values for hub/tip – leading to formation of fillets – minimum values for rest of the sections) and increased exit blade angles β_E (less flow turning) result in $\omega_0^* = 0.0731$ for OPS₀.

On the other hand, the exit flow angle deviation from the axial direction rises to $\alpha_{E,0}^* = 6.7^\circ$, which is mainly driven by its circumferential/whirl component α_w . The whirl angle distribution at the exit plane of the CFD domain is plotted in Fig. 10 for the BS (a) and OPS₀ (b) blades, and is primarily affected by the modifications in β_E .

ii.3 Pareto tracing

Once the first Pareto point has been computed, the proposed methodology following Algorithm 1 is employed, while requesting for 10 steps ($n_s = 10$) each of which attempts to reduce α_E by 0.5° ($\Delta\hat{\alpha}_E = -0.5^\circ$).

Figure 7 shows the 11 Pareto optimal points which were obtained in the non-dimensional objective space. These non-dominated solutions are in general well distributed along the Pareto front, apart from the final point \mathbf{b}_{10}^* which is further away from the previous point \mathbf{b}_9^* due to the "flattening" of the front. An intermediate point could be found (if that were of interest) by decreasing the step size $\Delta\hat{\alpha}_E$. The predicted points are close to the optimal solutions and, thus, are omitted from the plot (similar to the target points) for the sake of clarity. The point represented as a red triangle was obtained (at a preliminary stage) by performing an additional "Go-to-Pareto" optimization with $[w_1 \ w_2] = [0 \ 1]$, and seems to be in line with the shape of the computed Pareto front.

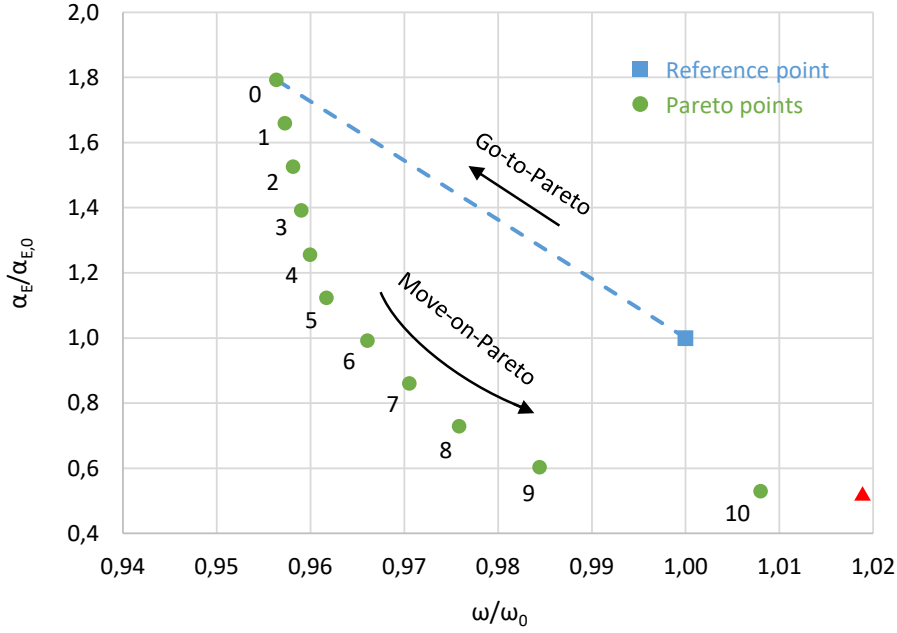


Figure 7: Bi-objective optimization of a 3D compressor stator: Pareto front in non-dimensional objective space approximated by 11 well-distributed optimal points obtained using the proposed methodology. The red filled triangle stands for an additional Pareto point, which resulted from a "Go-to-Pareto" optimization with $[w_1 \ w_2] = [0 \ 1]$

In terms of computational cost, the "Move-on-Pareto" process required in total 159 EFS for the primal and adjoint simulations (including line search steps). This consists of 21 EFS for the first step (without prediction) plus ~ 15 EFS on average for each subsequent step (using prediction-correction), which is considered to be efficient when compared to the 36 EFS needed for the "Go-to-Pareto" step. A summary of this cost comparison between the components of the method is tabulated in Table 1, where the benefit of incorporating a prediction step is shown.

All the Pareto points in Fig. 7 share minimum values for the design parameters r_I , r_E and T_{max} of all blade sections excluding hub and tip. The continuous geometry modifications when moving from point to point are attributed to variations in the blade angles β_I , β_E (controlling the loading of the blade), as well as to the 3D stacking of the sections (axial and circumferential shift parameters).

Table 1: Summary of computational cost to compute 11 Pareto points

	Total cost [EFS]	Cost per point [EFS]	BFGS steps per point
Go-to-Pareto	36	36	8
Move-on-Pareto (first step)	21	21	5
Move-on-Pareto (steps 2–10)	138	15.3	3.1

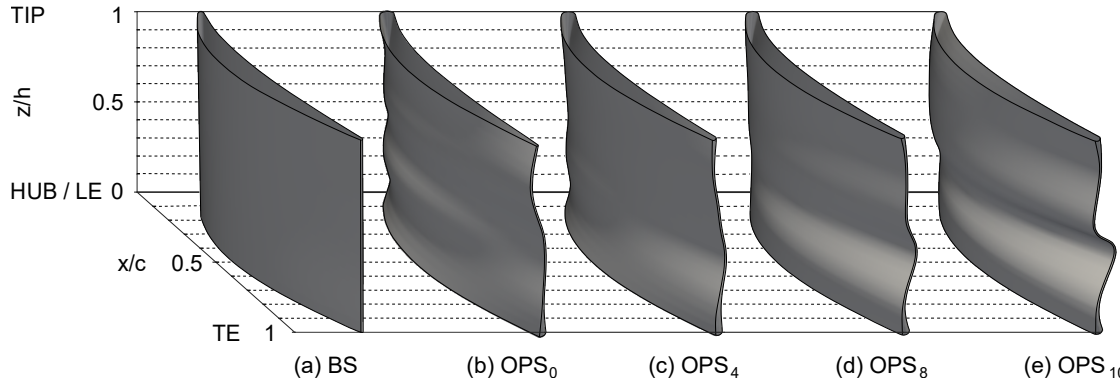


Figure 8: Comparison of stator shapes: (a) Blade for reference point \mathbf{b}_0 ; (b) Blade for initial Pareto point \mathbf{b}_0^* ; (c) Blade for Pareto point \mathbf{b}_4^* ; (d) Blade for Pareto point \mathbf{b}_8^* ; (e) Blade for final Pareto point \mathbf{b}_{10}^*

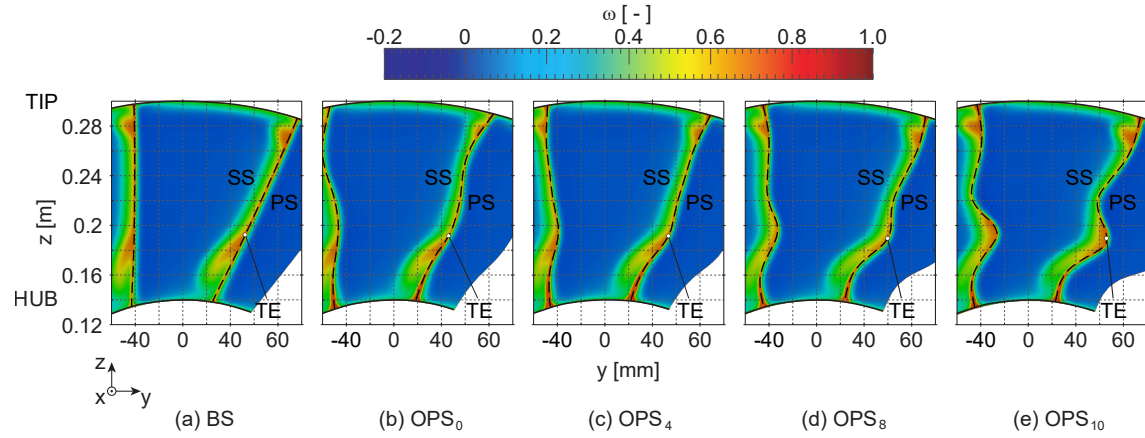


Figure 9: Total pressure loss coefficient distribution at the trailing edge plane. Contour plots for: (a) Reference point \mathbf{b}_0 ; (b) Initial Pareto point \mathbf{b}_0^* ; (c) Pareto point \mathbf{b}_4^* ; (d) Pareto point \mathbf{b}_8^* ; (e) Final Pareto point \mathbf{b}_{10}^*

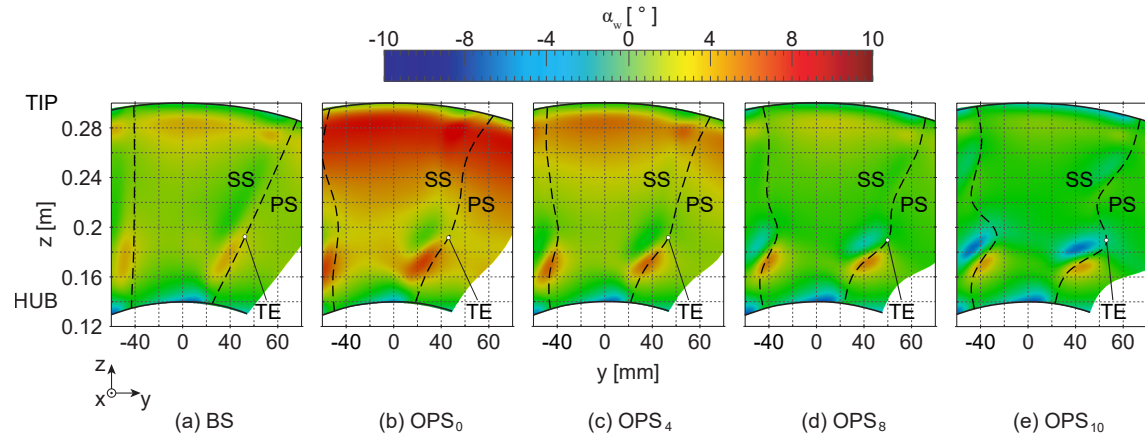


Figure 10: Whirl angle distribution at the exit plane. Contour plots for: (a) Reference point \mathbf{b}_0 ; (b) Initial Pareto point \mathbf{b}_0^* ; (c) Pareto point \mathbf{b}_4^* ; (d) Pareto point \mathbf{b}_8^* ; (e) Final Pareto point \mathbf{b}_{10}^*

This is illustrated in Fig. 8, where the geometries corresponding to the Pareto points \mathbf{b}_4^* (c), \mathbf{b}_8^* (d) and \mathbf{b}_{10}^* (e) are compared to the BS (a) blade. The progressively formed wavy TE is mainly responsible for reducing the exit flow angle to a minimum of $\alpha_{E,10}^* = 1.98^\circ$, as shown in Fig. 10, where the distribution of the whirl angle component at the exit plane of the CFD domain is plotted for the OPS₄ (c), OPS₈ (d) and OPS₁₀ (e) blades. The OPS₁₀ blade achieves an average whirl angle closer to zero.

Finally, the total pressure loss coefficient distribution at an axial cross-section just behind the TE is plotted in Fig. 9 for all the discussed blades. The one corresponding to OPS₁₀ (e) is slightly worse than the BS one (a) and results in $\omega_{10}^* = 0.077$.

Performing the presented MOO is clearly beneficial to the design process, since it simplifies the task of the design engineer, who can select one of the computed non-dominated solutions (e.g. the OPS₈ which dominates the BS w.r.t. both objectives: $\omega_8^* = 0.0746$, $\alpha_{E,8}^* = 2.73^\circ$) according to his/her preference.

IV. CONCLUSIONS

In this paper, a new adjoint-based method for the solution of bi-objective optimization problems was developed. The method relies on an iterative prediction-correction approach for efficiently tracing the Pareto front. The main difference from other curve continuation techniques exists in the formulation of the KKT conditions, where a distance function is treated as the objective. A BFGS optimization method was implemented in order to approximate the Hessian information required at each prediction step of the algorithm.

The proposed method was first demonstrated in the lift/drag optimization of an isolated airfoil, and then successfully applied to the bi-objective optimization of a 3D compressor stator. It was found that the BFGS Hessian approximations can further be utilized to accelerate the "Move-on-Pareto" technique. Here, the introduction of the prediction step seems to increase method's efficiency by a factor of ~ 1.4 (average value for examined cases). As a result, the presented Pareto fronts were obtained in an automatic manner with low computational cost and well-distributed solutions. The extension of the method to multidimensional MOO problems is straightforward, provided that an algorithm to set the targets (for sweeping the Pareto front) is found.

ACKNOWLEDGEMENTS

The first author was an ESR of the IODA project [39], funded by the European Union HORIZON 2020 Framework Programme for Research and Innovation under Grant Agreement No. 642959, during the period 2015-2018. In addition, the authors would like to thank Rolls-Royce Deutschland for the permission to publish the results regarding the turbomachinery application.

This is a post-peer-review, pre-copyedit version of an article published in Engineering with Computers. The final authenticated version is available online at: <http://dx.doi.org/10.1007/s00366-019-00832-y>.

REFERENCES

- [1] Nocedal J, Wright S (2006) Numerical Optimization. Springer-Verlag, New York
- [2] Deb K (2001) Multi-objective Optimization using Evolutionary Algorithms. John Wiley & Sons Inc., New York

- [3] Marler RT, Arora JS (2004) Survey of Multi-Objective Optimization Methods for Engineering. *Structural and Multidisciplinary Optimization* 26(6):369–395
- [4] Zingg DW, Nemec M, Pulliam TH (2008) A comparative evaluation of generic and gradient-based algorithms applied to aerodynamic optimization. *European Journal of Computational Mechanics* 17(1-2):103–126
- [5] Kyriacou SA, Asouti VG, Giannakoglou KC (2014) Efficient PCA-driven EAs and metamodel-assisted EAs, with applications in turbomachinery. *Engineering Optimization* 46(7):895–911
- [6] Das I, Dennis JE (1998) Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems. *SIAM Journal Optimization* 8(3):631–657
- [7] Kim IY, de Weck OL (2005) Adaptive weighted-sum method for bi-objective optimization: Pareto front generation. *Structural and Multidisciplinary Optimization* 29(2):149–158
- [8] Mueller-Gritschneider D, Graeb H, Schlichtmann U (2009) A Successive Approach to Compute the Bounded Pareto Front of Practical Multiobjective Optimization Problems. *SIAM Journal Optimization* 20(2):915–934
- [9] Gebken B, Peitz S, Dellnitz M (2019) On the hierarchical structure of Pareto critical sets. *Journal of Global Optimization* 73(4):891–913
- [10] Shankaran S, Barr B (2011) Efficient Gradient-based Algorithms for the Construction of Pareto Fronts. In: *ASME Turbo Expo 2011*. ASME Paper GT2011-45069, Vancouver, Canada
- [11] Fike JA (2013) Multi-Objective Optimization using Hyper-Dual Numbers. PhD thesis, Stanford University, USA
- [12] Peitz S, Ober-Blöbaum S, Dellnitz M (2018) Multiobjective Optimal Control Methods for the Navier-Stokes Equations Using Reduced Order Modeling. In: *Acta Applicandae Mathematicae*. Springer, Netherlands, pp 1–29
- [13] Banholzer S, Beermann D, Volkwein S (2017) POD-Based Error Control for Reduced-Order Bicriterial PDE-Constrained Optimization. *Annual Reviews in Control* 44:226–237
- [14] Schmidt S, Schulz VH (2008) Pareto-curve continuation in multi-objective optimization. *Pacific Journal of Optimization* 4(2):243–257
- [15] Gkaragkounis K, Papoutsis-Kiachagias E, Asouti V, Giannakoglou K (2018) Adjoint-based Pareto Front Tracing in Aerodynamic Shape Optimization. In: *10th International Conference on Computational Fluid Dynamics (ICCFD10)*, Barcelona, Spain
- [16] Papadimitriou DI, Giannakoglou KC (2012) Aerodynamic design using the truncated newton algorithm and the continuous adjoint approach. *International Journal for Numerical Methods in Fluids* 68(6):724–739
- [17] Tsiakas KT, Trompoukis XS, Asouti VG, Giannakoglou KC (2018) Shape Optimization of Wind Turbine Blades Using the Continuous Adjoint Method and Volumetric NURBS on a GPU Cluster. In: *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*. Springer, pp 131–144

- [18] Giles MB (2000) On the use of Runge-Kutta time-marching and multigrid for the solution of steady adjoint equations. In: AD2000 Conference, Nice, France
- [19] Ehrgott M (2005) Multicriteria Optimization. Springer-Verlag, Berlin/Heidelberg
- [20] Piegel L, Tiller W (1997) The NURBS Book. Springer-Verlag, Berlin
- [21] Gagliardi F, Tsiakas KT, Giannakoglou KC (2017) A Two-Step Mesh Adaptation Tool Based on RBF with Application to Turbomachinery Optimization Loops. In: EUROGEN International Conference 2017, Madrid, Spain
- [22] Asouti VG, Trompoukis XS, Kampolis IC, Giannakoglou KC (2011) Unsteady CFD computations using vertex-centered finite volumes for unstructured grids on Graphics Processing Units. International Journal for Numerical Methods in Fluids 67(2):232–246
- [23] Spalart PR, Allmaras SR (1992) A one-equation turbulence model for aerodynamic flows. In: AIAA 30th Aerospace Sciences Meeting and Exhibit, Reno, USA
- [24] EASY: The Evolutionary Algorithms SYstem. <http://velos0.ltt.mech.ntua.gr/EASY/>. Accessed 24 April 2019
- [25] Vasilopoulos I, Flassig P, Meyer M (2017) CAD-based Aerodynamic Optimization of a Compressor Stator using Conventional and Adjoint-driven Approaches. In: ASME Turbo Expo 2017. ASME Paper GT2017-63199, Charlotte, NC, USA
- [26] Müller JD (2016) TurboLab Stator. <http://aboutflow.sems.qmul.ac.uk/events/munich2016/benchmark/testcase3/>. Accessed 26 April 2019
- [27] Gräsel J, Keskin A, Swoboda M, Przewozny H, Saxer A (2004) A Full Parametric Model for Turbomachinery Blade Design and Optimisation. In: ASME 2004 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. ASME Paper DETC2004-57467, Salt Lake City, Utah
- [28] Dutta AK (2011) An Automated Multi-Objective Optimization Approach for Aerodynamic 3D Compressor Blade Design. Dissertation, Brandenburg University of Technology Cottbus-Senftenberg, Germany
- [29] Abbott IH, Doenhoff AE (1960) Theory of Wing Sections, Including a Summary of Airfoil Data. Dover Publications Inc., New York
- [30] Shahpar S, Lapworth L (2003) PADRAM: Parametric design and rapid meshing system for turbomachinery optimisation. In: ASME Turbo Expo 2003. ASME Paper GT2003-38698, Atlanta, Georgia
- [31] Launder BE, Spalding DB (1974) The numerical computation of turbulent flows. Computer Methods in Applied Mechanics and Engineering 3(2):269–289
- [32] Lapworth BL (2004) HYDRA-CFD: A Framework for Collaborative CFD Development. In: International Conference on Scientific and Engineering Computation (IC-SEC) 2004, Singapore
- [33] Shahpar S, Caloni S (2013) Aerodynamic optimization of high-pressure turbines for lean-burn combustion system. Journal of Engineering for Gas Turbines and Power 135(5), 055001
- [34] Moinier P (1999) Algorithm Developments for an Unstructured Viscous Flow Solver. PhD thesis, University of Oxford, UK

- [35] Moinier P, Müller JD, Giles MB (2002) Edge-based multigrid and preconditioning for hybrid grids. *AIAA Journal* 40(10):1954–1960
- [36] Mihalyovics J, Brück C, Vasilopoulos I, Meyer M (2018) Numerical and Experimental Investigations on Optimized 3D Compressor Airfoils. In: *ASME Turbo Expo 2018*. ASME Paper GT2018-76826, Lillestrøm (Oslo), Norway
- [37] Vasilopoulos I, Agarwal D, Meyer M, Robinson TT, Armstrong CG (2016) Linking Parametric CAD with Adjoint Surface Sensitivities. In: *ECCOMAS Congress 2016*, Crete Island, Greece
- [38] Xu S, Radford D, Meyer M, Müller JD (2015) CAD-based adjoint shape optimisation of a one-stage turbine with geometric constraints. In: *ASME Turbo Expo 2015*. ASME Paper GT2015-42237, Montreal, Canada
- [39] Initial Training Network (2015) Industrial Optimal Design using Adjoint CFD. <http://ioda.sems.qmul.ac.uk/>. Accessed 26 April 2019