Airfoil Optimization using Design-by-Morphing

We present Design-by-Morphing (DbM), a novel design methodology to create a search space for topology optimization of 2D airfoils. Most design techniques impose geometric constraints or designers' bias on the design space itself, thus restricting the novelty of the designs created, and only allowing for small local changes. We show that DbM methodology doesn’t impose any such restrictions on the design space, and allows for extrapolation from the search space, thus allowing for truly radical and large search space with only a few parameters. We apply DbM to create a search space for 2D airfoils, and optimize this shape design space for maximizing the lift-over-drag ratio, $CL_{max}$, and stall angle tolerance, $\Delta \alpha$. Using a genetic algorithm to optimize the DbM space, we show that we create a Pareto-front of radical airfoils that exhibit remarkable properties for both objectives.

Keywords: Design-by-Morphing (DbM), Topology Optimization, Airfoils

1 Introduction

Optimizing the shape of an airfoil is an integral design stage for aerodynamic components like aircraft wings[1–4] and wind-turbine blades[5–10]. A typical airfoil optimization process contains three main components: shape parameterization, airfoil evaluation, and optimization, among which the parameterization method determines both the design space and the complexity of the optimization problem. In this sense, a desirable parameterization technique must cover a wide design space within a limited number of design parameters[11–13], which is especially important during the early design stage when minimum geometric constraints are placed and radical changes during the optimization process are welcomed.

Different shape parameterization methods offer different fidelity and ranges of control[11,13,14], and, according to the scope of the design parameters, one can place these methods on a spectrum where, on one end, the change of one parameter affects only a local area of the airfoil surface thus altering a finer control of the shape, and, on the other end, each parameter affects the airfoil’s global contour[11].

On the local end of the spectrum is the discrete method[15], whose design parameters are exactly the discrete surface points that define the airfoil shapes. Because the displacement of each point can be adjusted, the design space is potentially limitless[16], and very fine local control as well as high fidelity can be achieved. However, to describe an airfoil shape accurately, a large number of surface points are needed, which increases the complexity of the optimization problem. Furthermore, to accommodate the large number of design variables, one usually uses gradient-based method to guide the optimization which is limited to small local changes and can easily get stuck at a local optimum.

Increasing the geometrical extent of each parameter’s influence, one would find classical methods that determine the airfoil shape based on the regional features or the control points and perform curve-fittings of some kind. For example, the popular parametric section (PARSEC) method[17] uses eleven parameters that represent specific sectional features such as leading edge radius and upper and lower crest locations and approximate the airfoil surface using a 6th order polynomial. Another popular method would be the Bézier parameterization[18], which forms the upper and the lower surfaces of the airfoil through the Bézier curves defined by the pre-selected control points. Additionally, a combination of the two techniques, Bézier-PARSEC parameterization[19], also exists, which creates Bézier curves using the parameters of the PARSEC method and combines these curves to form the shape contours. One main issue with the above methods is their inability or inefficiency to include high-fidelity features: the PARSEC and the Bézier-PARSEC method have fixed number of parameters and offer very limited range of fidelity, while the Bézier parameterization requires higher-degree Bézier curves to describe complex shapes, which become inefficient to calculate as the order increases[16].

To include high-fidelity features, or, equivalently, represent more complex curves, B-splines[20,21], including nonuniform rational B-spline (NURBS)[22], can be used, which form curves by connecting low-order Bézier segments defined by the control points. Naturally, with denser control points, these methods move to the local end of the spectrum and are able to represent high-fidelity features, but the computation complexity also increases. In an effort to reduce the number of the design parameters, the control points can be grouped together, and global transformations such as twisting and thickening can be used as the parameters instead. This is known as the free-form deformation (FFD) method[23,24] and is closer to the global end of the spectrum. A similar method, namely Radial Basis Function Domain Element (RBF) approach[25,26], also exists and uses radial basis function to exert deformation on the airfoil.

To the global end of the spectrum includes methods that use spectral construction of some basis functions or modes to form or deform the airfoil shapes. One typical way of determining the basis modes is through proper orthogonal decomposition (POD) of a set of airfoil data, and dimensionality can be reduced by using only the dominant modes[27,28]. Other methods include the Hicks-Henne’s approach[29], which uses a linear combination of sines functions to deform the airfoil surface, and class/shape function transformation (CST) method proposed by Kulfan[30,
by the baseline shapes, which can create truly novel and unusual 'interpolation' between the shapes, applying negative weights during morphing allows ‘extrapolation’ from the search space spanned by the baseline shapes, which can create truly novel and unusual shapes. Lastly, DbM is completely free from any geometric parameter constraints and the only implicit constraints are the selections of the ‘baselines shapes’ themselves.

For 2D airfoils, the closed shapes can be collocated in the Euclidean coordinate system. We note here that all 2D shapes bounded by a single surface are homeomorphic to one another. Using the leading edge of each airfoil as origin, each shape can be collocated by taking fixed and uniformly spaced points on the x-axis, which creates a one-to-one correspondence between the shapes. This collocation strategy is demonstrated in Figure 1, and the baseline shapes used in this paper are chosen from various airfoils from literature, which are detailed later. Morphing is performed by multiplying a specific airfoil shape with a scalar weight, summing up the weighted vectors, and then normalizing them, and thereby creating a one-to-one correspondence between the shapes via some geometric space[37,38]. The new shapes can then be generated systematically in either the functional[36] or geometric space. In order to be ‘morphed’ together, these baseline shapes used in this paper are chosen from various pre-existing designs in the literature, for the design search space exploration. However, this study, like many other dimension reduction methods, relies on the assumption that the optimum design is not far from an existing database, which is not always true. For our study, we mainly consider the shape parameterization technique for the early design stage and prefer not to make the same assumption. In particular, We are interested in a method that would contain high-order features while keeping a limited number of design parameters and allowing radical change from the initial airfoil shapes.

In this paper, we apply the Design-by-Morphing (DbM) parameterization technique, a novel design strategy that was introduced by Oh et al.[36] and has been used in recent years for geometry optimization of different problems[36–38], to the airfoil optimization problem. Specifically, our DbM method ‘morphs’ the baseline shapes together to create new shapes and can interpolate as well as extrapolate the design space, which allows for both the high-fidelity representation of shapes without the curse of dimensionality and radical improvements in the shapes without any geometric constraints[36,38]. Our paper makes the following scientific contributions:

- A DbM parameterization technique designed for the two-dimensional airfoil shape optimization allowing both accurate reconstruction of the existing airfoil database and radical change of airfoil shapes while being free of geometric constraints and designers’ biases.
- An optimization strategy using the DbM parameterization technique and the genetic algorithm that is able to create the Pareto-front of multi-objective airfoil optimizations.

2 Design-by-Morphing

Design-by-Morphing (DbM) works by morphing homeomorphic, i.e. topologically equivalent, shapes to create a continuous and constraint-free design search space that can produce radical extrapolated shapes, something which is unique from existing design strategies. The details of DbM are presented in the subsequent subsections.

2.1 Baseline Shapes and Morphing. The DbM technique generally requires two or more ‘baseline shapes’, usually chosen from pre-existing designs in the literature, for the design search space creation. In order to be ‘morphed’ together, these baseline shapes must be homeomorphic, which can be achieved by establishing a one-to-one correspondence between the shapes via some systematic shape collocation methods in either the functional[36] or geometric space[37,38]. The new shapes can then be generated by applying weights to the baseline shape collocation vectors and summing them together.

The DbM method is valid for shapes of any dimensions, and because radically different baseline shapes can be morphed together, exotic shapes can be created. Furthermore, in addition to ‘interpolation’ between the shapes, applying negative weights during morphing allows ‘extrapolation’ from the search space spanned by the baseline shapes, which can create truly novel and unusual shapes. Lastly, DbM is completely free from any geometric parameter constraints and the only implicit constraints are the selections of the ‘baselines shapes’ themselves.

For 2D airfoils, the closed shapes can be collocated in the Euclidean coordinate system. We note here that all 2D shapes bounded by a single surface are homeomorphic to one another. Using the leading edge of each airfoil as origin, each shape can be collocated by taking fixed and uniformly spaced points on the x-axis, which creates a one-to-one correspondence between the shapes. This collocation strategy is demonstrated in Figure 1, and the baseline shapes used in this paper are chosen from various airfoils from literature, which are detailed later. Morphing is performed by multiplying a specific airfoil shape with a scalar weight, summing up the weighted vectors, and then normalizing them, which is given by Eq. 1:

\[
P(x) = \frac{1}{2^\infty} \sum_{m=1}^{M} \sum_{n=1}^{N} w_n S_n(x).
\]  

Here \(S_n(x)\) is the y-coordinate collocation vector determining the \(n\)th baseline shape, collocated at \(x = [x_0, \ldots, x_F]\) where the \(i\)th x-coordinate \(x_i = [1 - 2i/F] \) and \(F\) is the number of collocation points. Accordingly, the first half elements of \(S_n\) represents the top surface of the airfoil and the second half elements of \(S_n\) renders the bottom surface of the airfoil. \(N\) is equal to the total number of baseline shapes. \(w_n \in [-1, 1]\) is the morphing weight applied to the y-coordinate vector of the \(n\)th baseline shape. A visual demonstration of our strategy is presented in Figure 2.
To ensure diversity and to introduce radical features in the design of airfoils, we select baseline shapes plotted as a function of the index $i$ of the collocation vector. The airfoils are then weighted to the corresponding baseline shapes. Our results in later processes[39], we deliberately include the bad performers so that our optimization can suppress their features by assigning negative weights to the corresponding baseline shapes. Our results in later sections will demonstrate this in greater detail.

### 2.2 Intersection Control.
For smooth baseline shapes, applying positive weights, i.e. interpolation, will always create smooth shapes without applying any geometric constraint. However, because the DbM imposes no geometric parameter restraints, extrapolation, i.e. applying negative weights, may produce non-physical geometries such as self-intersections, which have ‘zero-area’ regions as shown in Figure 3(a). One may discard the morphed airfoil shapes with self-intersections during the optimization but that diminishes the size of our design space. Instead, we recover new shapes by removing the intersections. This is accomplished by first locating within the morphed coordinate vector where the intersection occurs and restructuring the coordinate vector by ‘flipping’ it between the intersection points as shown in Figure 3(c). The vector is then ‘stiffened’ to remove the zero area between the intersections by removing the points in their neighborhoods and then linearly interpolating between the broken coordinate vectors. As seen in Figure 3(d), this removes the ‘zero-area’ space and gives some physical area to the shape at the point of intersection. The above process is repeated until all intersections are removed, e.g. both intersections in Figure 3 are successfully removed, and, finally, a moving-average smoothing filter is applied to smooth out the sharp edges.

### 2.3 Baseline Shape Selection.
The selection of baseline shapes is an important component of DbM strategy and ultimately determines the size and the novelty of our search space. Metaphorically, the selection of the baseline airfoil shapes serves as the gene pool for the morphed airfoils, and its diversity is important for creating a large design space. Our baseline shape selection contains good ones with either high lift-to-drag ratio or good stall performance, bad ones with poor aerodynamic performance, commonly used airfoil shapes in the literature or industry, and airfoils with irregular shapes to provide novelty to the design space. It is worth noting that, contrary to the conventional airfoil optimization processes[39], we deliberately include the bad performers so that our optimization can suppress their features by assigning negative weights to the corresponding baseline shapes. Our results in later sections will demonstrate this in greater detail.

In this paper, we select 25 baseline shapes (see Figure 4) from the UIUC airfoil coordinates database[40]. The airfoils are selected to ensure diversity and to introduce radical features in the design space. Their model names and characteristics are attached in Appendix B. Each airfoil shape is represented by 4000 coordinates that span from the first surface trailing edge around the leading edge to the second surface trailing edge with equally distributed $x$-coordinates parallel to the airfoil chord line of a unit length.

### 2.4 Representation Capacity.
To examine the robustness and the extent of our design space generated by the morphing of only 25 airfoil baselines, we reconstructed the pre-existing 2D airfoil shapes archived in the UIUC airfoil database[40] via DbM. A total of 184 randomly-chosen airfoils are tested, which accounts for approximately 11% of the UIUC database as of 2022. All 184 airfoils were reconstructed to the same accuracy, and a random selection of 100 out of the 184 airfoils are shown in Figure 5.

For each airfoil, the shape was reconstructed by running a global optimization of the weight vector that minimizes the total area of the original and morphed shapes where one shape does not overlap with the other, e.g. the geometric XOR of 2 closed shapes. Note that the geometric XOR serves as a good measure of the similarity between 2 shapes since it gradually goes to zero as the shapes become identical to one another. As a result, all the test airfoils were successfully re-created by the morphing of only 25 baseline shapes with an area difference of less than 0.01. It means the average error in $y$-coordinate is <0.5% since the airfoil chord is normalized as a unit length and the area is formed by 2 airfoil surfaces. This affirms that the current 25 baseline shapes are diverse enough to span the design space via DbM to explore airfoils in a universal manner.

Overall, by using only 25 dimensions, our DbM method is able to recreate the UIUC database airfoils with high fidelity, and it does not suffer from the ‘curse of the dimensionality’ compared to other techniques where the fidelity of the parametrization depends on the number of independent dimensions used. Moreover, DbM’s capability of creating exotic shapes via its extrapolation feature increases the chance to find novel solutions that are deviated from the previously-established space like the UIUC database. This exploration is essential especially for the airfoil design where the correlation between the geometric feature and the aerodynamic performance of an airfoil can be very non-intuitive, thus necessitating exploratory design spaces.
and then smoothed over, shown by hat coordinates removed by linear interpolation to remove the intersecting area shapes and then evaluates and optimizes the airfoils formed by technique introduced in Sec. 2. As shown by the flowchart in 3 Optimization Methodology

\[ f_2(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ \frac{x}{10} & \text{if } 0 < x \leq 10 \\ \frac{1}{x} & \text{if } x > 10 \end{cases} \]

4 Optimization Methodology

Our airfoil optimization methodology is built around the DbM technique introduced in Sec. 2. As shown by the flowchart in Figure 6, the optimization starts from the selection of the baseline shapes and then evaluates and optimizes the airfoils formed by morphing these baseline shapes using DbM. Our methodology does not rely on one specific airfoil evaluation tool or one specific optimizer, and discussions on their choices are provided in Sec. 3.1 and Sec. 3.2 respectively.

3.1 Airfoil Evaluation. Our optimization methodology is not limited to one particular airfoil performance analyzer. One can use any CFD or experimental methods. For the optimization of airfoil shapes using CFD-based solvers, the evaluation of the objective functions (aerodynamic properties) typically falls into two categories: the full Reynolds-averaged Navier-Stokes (RANS) based approach and the interacted viscous/inviscid zonal approach. The RANS-based approach is computationally expensive and demands the optimizer to be highly efficient, and, to accommodate the large number of design variables as often seen in the aerodynamic designs, gradient-based optimizers coupled with adjoint methods for computing the derivatives are deemed the most feasible[41–43].

On the other hand, the viscous/inviscid zonal approach, which combines separate solutions for the inviscid external flow and the viscous shear layer flow iteratively to form a continuous profile, is faster and less expensive. Among the many inviscid/viscous airfoil analysis codes, the XFOIL program[44] has been the most dominant and widely adopted[45–52]. It couples a vorticity panel method for exterior flow with an integral boundary-layer method for viscous boundary layers and uses an $e^k$ type amplification formula to determine the transition point[44]. Its applicability to airfoil designs has been demonstrated in the past literature, where its predictions of aerodynamic properties are shown in good agreement with the wind-tunnel experiment data[53,54] and the RANS-based simulation results[55].

The specific choice of the evaluation tool used in this paper is not essential to manifest the power of the DbM parameterization technique, which is the main focus of our paper. For this work, we opt for XFOIL because of its acceptable accuracy under our flow condition as well as its low computation cost. Its wide usage also allows quick reproduction of our optimization results. It is used in a black-box manner so that any other commercial or in-house airfoil analysis tools can be incorporated into our optimization framework if necessary. Our detailed airfoil evaluation setup is attached in the appendix B.

3.2 Optimization. When given a set of solutions, for single objective optimization problems, the most optimal solution within the set can be determined. However, for multi-objective optimization, multiple and potentially conflicting objectives must be considered simultaneously to determine the optimal answer in the solution set[56,57]. If the designer has a quantitative ranking of the objectives, these objectives can be combined together to formulate a single objective problem, but when no such ranking exists, constructing a Pareto front is the most common methodology[58–60], which is applicable to real-world problems such as the design of architected materials[61,62], turbo-machinery[63–67], process-engineering[68–70], shape design[71–73], and structural engineering[74,75] when multiple objectives that cannot be quantitatively ranked are involved.

We pose the multi-objective optimization problem as

\[ w_{opt} = \arg\max_{w} (f(w)), \]

where $f(w) = [f_1(w), f_2(w), \ldots, f_k(w)]$. Here $f_1, \ldots, f_k$ are the $k$ objectives to be maximized and $w$ is the design variable vector. Generally $w$ is a $d$-dimensional vector defined over a bounded set $W \subset \mathbb{R}^d$ representing $d$ continuous variables. \(\{w_{opt}\}\) is a set of Pareto-optimal solution vectors, i.e., a vector which is not Pareto-dominated by any other vector. For the reader’s convenience, it is noted that a design variable vector $\hat{w}$ is Pareto-dominated by another design variable vector $\tilde{w}$ if $f_k(\hat{w}) \leq f_k(\tilde{w})$ for all

![Figure 3](image1.png)

Fig. 3 Conditioning for intersection removal. (a) Intersections are detected; (b) Blown up image of one intersection. Shape coordinates direction is depicted by arrows; (c) Intersection removed by flipping vector between intersection; (d) Zero area removed by linear interpolation to remove the intersecting area and then smoothed over, shown by hat coordinates

![Figure 4](image2.png)

Fig. 4 Twenty-five baseline shapes picked from the UIUC airfoil coordinates database[40]. See Appendix B for more details.
To obtain the Pareto-front, especially when objectives cannot be weighted or when a non-convex black-box function is considered, evolutionary or genetic algorithms are a natural choice[70,76]. In fact, they have been commonly implemented in many previous aerodynamic optimization studies due to their gradient-free nature and wide region of the search domain[31,77–79]. On the other hand, when the cost functions are expensive to compute (e.g. when using experiments as an evaluation tool), Bayesian optimization methods have proven to be efficient[80].

Our study considers a bi-objective \( K = 2 \) two-dimensional airfoil shape optimization. In particular, we optimize the shape of a subsonic airfoil operating in an incompressible flow with \( \text{Re} \equiv \frac{U c}{\nu} \times 10^6 \), where \( U \) and \( \nu \) are the free-stream flow speed and fluid kinematic viscosity, respectively, and \( c \) is the airfoil chord length. The parameter to be optimized is the morphing weight vector for the DbM technique:

\[
\mathbf{w} \equiv (w_1, \ldots, w_{25}) \in \mathcal{D}^{25},
\]

where \( \mathcal{D} = [-1, 1] \subset \mathbb{R} \) and \( w_i (i = 1, 2, \ldots, 25) \) is the weight applied to the \( i \)-th baseline shape. The design objectives to be maximized are the maximum lift-drag ratio at any angle of attack \( \alpha \), i.e. \( f_1(\mathbf{w}) = CLD_{\text{max}}(\mathbf{w}) \), and the difference between the stall angle \( \alpha_s \) and the angle where the maximum lift-drag ratio occurs, i.e. \( f_2(\mathbf{w}) = \Delta \alpha(\mathbf{w}) \), where \( \Delta \alpha \) is often called the stall angle tolerance. Precise definitions of these design objectives are explicated in Appendix A, and both objectives are evaluated using the XFOIL simulations, which are efficient enough to be used with the genetic algorithm.

We use a MATLAB-based variant of the popular NSGA-II[81] algorithm, which is a controlled, elitist genetic algorithm. Our initial population consists of the single-objective optimums of each design target as well as random samples in the design space. The population size of 372 is used with a total of 3,000 maximum generations, and the solutions are actively ranked within each generation so as to maintain diversity and avoid over-crowding in the Pareto-optimal solution set. Our setup was tested on the commonly used set of ‘ZDT’ benchmark problems for multi-objective problems, suggested by Zeidler et al.[82]. The details of the test problems and the validation results are provided in Appendix C.

4 Results

The Pareto front on the \( \Delta \alpha - CLD_{\text{max}} \) objective plane, which resulted from the 3,000 generation genetic algorithm (GA) runs, is depicted in Figure 7. The convergence of the front is confirmed by the large generation number with the population size of 372, involving around 1.1 million XFOIL evaluations of \( CLD_{\text{max}} \) and \( \Delta \alpha \). After non-dominant or duplicate individuals are removed in the final generation of the population, we are able to identify 208 Pareto-dominant airfoil shapes composed via DbM using 25 baseline airfoil shapes. For comparison, these 25 baseline shape cases are evaluated and plotted as red hollow circles in Figure 7 together. The reason why baseline #19 has zero \( CLD_{\text{max}} \) and \( \Delta \alpha \) is that we inverted the shape intentionally and therefore XFOIL failed to evaluate its aerodynamic performance. We assigned the objective functions zero values for such failing cases because they represented...
airfoil geometries that are not aerodynamically viable in the XFOIL space. The GA optimization successfully developed the Pareto front, where two ends are posed at \((CLD_{\text{max}}, \Delta \alpha) = (30.63, 40^\circ)\) and \((CLD_{\text{max}}, \Delta \alpha) = (273.39, 10^\circ)\). Even in the largest maximum lift-drag ratio case, the angle of attack gap between stall and design point is found to be \(10^\circ\), giving the airfoil a tolerant range for off-design operations.

The front is divided into 3 different clusters, each of which constitutes a segment of the front which does not overlapping one another. It is worth noting that the non-overlapping division of the front is a consequence of clustering through the Principal Component Analysis (PCA), rather than arbitrary. The detail of the clustering is provided in Appendix D. Figure 8 shows nine representative optimal airfoil shapes on the Pareto front in ascending order of \(CLD_{\text{max}}\). In each cluster, three airfoil shapes having considerably different objective function values were selected to be presented. Also, note that Figure 8(a) illustrates the extreme case of the smallest \(CLD_{\text{max}}\) and largest \(\Delta \alpha\) while Figure 8(i) depicts the other extreme of the largest \(CLD_{\text{max}}\) and smallest \(\Delta \alpha\). It can be seen that within the cluster the overall shape remains identical and only a gradual decrease in the airfoil thickness is observed as \(CLD_{\text{max}}\) increases. Since thin airfoils such as bird-like airfoils[83], which we take as part of the baseline and smallest, while Figure 8 illustrates the extreme case of the smallest.

The mean weight distributions with respect to 25 original baseline shapes are given in Figure 9. Overall, the weight distributions of 3 clusters comply with the weight distribution of the total mean. It turned out that baseline shape #13 (model name: AS6097) is commonly the most significant one for morphing. Since this baseline shape is the best in \(CLD_{\text{max}}\) and the second best in \(\Delta \alpha\) among 25 baseline shapes (see Figure 7), it was likely to survive in the GA runs over the generations against the selection pressure that only sorts out dominant individuals in terms of both \(CLD_{\text{max}}\) and \(\Delta \alpha\). However, excellence in the objectives of an individual baseline shape does not necessarily guarantee its survival, which is the case for the globally best baseline shape #6 (model name: AH 79-100C), as an individual’s superior ‘phenotype’ may be no longer revealed, or even suppressed after the morphing is done and all ‘genes’ are mixed with each other.

As we discussed from the examination of the morphed airfoil shapes, both cluster 1 and 2’s mean weight distributions show no considerable difference from the total mean weight distribution. Through small shape variation from the total mean airfoil shape as in Figure 10(a), it is possible to reach these optima relatively easily. In contrast, cluster 3 has a number of weights that are quite different from the mean (e.g., #6 and #11) and substantial morphing would be required if one starts with the total mean airfoil shape.

In the context of the present study, each axis obtained by the PCA can be represented by a unique form of morphed airfoil shapes because 25 PCA coefficient vectors defined in the weight space \(D^{25}\) are orthogonal to each other. These 25 new morphed airfoils span the whole design space and therefore serve as alternative baseline shapes in lieu of the original ones. More importantly, the dominance of the first 2 PCA axes with respect to the data point variance suggests that the major geometric feature of 208 airfoil shapes we found via the optimization is virtually generated by morphing of these two new airfoils. Small variance of a PCA axis indicates that the data points are not considerably deviated from their mean on the axis. In other words, the baseline shape corresponding to this PCA axis has a marginal impact on morphing the airfoil shape for optimization. Once we pick two baseline shapes from the first two dominant PCA axes, whose associated collocation vectors are say \(P_1\) and \(P_2\), and use them to morph the airfoil shape obtained from the total mean of the Pareto-optimal weight vector set, which corresponds to the mean collocation vector \(P_{\text{mean}}\), we get better understanding of how the morphing, especially along each PCA axis, has an influence on major geometric changes in the optimal airfoil shapes. These airfoil shapes are depicted in Figure 10, where the black and red surfaces are generated from the first and second half of the collocation points, respectively. For example, we note that the orientation of two surface of \(P_1\) is flipped in comparison to that of \(P_{\text{mean}}\), meaning that the stronger the weight of PCA axis 1 in the positive direction is, the narrower a morphed airfoil shape is.
Fig. 8 Nine representative Pareto-optimal airfoil shapes. (a)-(c) are in cluster 1, (d)-(f) are in cluster 2 and (g)-(i) are in cluster 3.

5 Discussion

Most parameterization strategies depend upon careful selection of constraints and parameters, which determines their probability of success. The fidelity offered by such methods is very dependent on the number of the parameters chosen. Moreover, these designs are limited by the parametric constraints and the implicit designer’s bias, making extrapolation or radical global changes difficult. Data driven methods typically rely on the assumption that the optimum solutions are not far from the training data-set, which again prevents radical shape changes.

Design-by-Morphing, on the other hand, creates a design space that is uninhibited by any geometric constraints and also allows extrapolation from the design space. It doesn’t suffer from the curse of dimensionality when parameterizing airfoils and allows high-fidelity representation of airfoils without increasing the number of independent parameters in the problem. Using only 25 baseline shapes from the UIUC database, we were able to recreate the UIUC database with 0.5% error. We also showed that radical, global changes are possible using DbM. Applying that for the bi-objective shape optimization with objectives of maximizing $CL_{D_{\text{max}}}$ and $\Delta \alpha$, we were able to achieve significant results compared to our baseline shapes.

We posit that for design parametrization of airfoils and for other 2D/3D shapes, DbM should be the method of choice for creating an unconstrained, unbiased and non-data intensive design space that allows radical modifications, which can often be non-intuitive shapes.

6 Conclusion

DbM methodology creates a design space for radical 2D airfoils. We show that the space creates novel airfoils that are not constrained by geometric parameters or designer bias. Optimizing the design space created for dual objectives of $CLD$ and $\Delta \alpha$, we show remarkable improvements in both objectives and provide a Pareto-front of optimal airfoil designs. Our final airfoils show remarkable improvements over our existing baseline shapes. For optimizing 2D or 3D airfoils, DbM should be used as the method of choice for design space creation. Moreover, our methodology is flexible to be used for optimizing shapes for other fluid machinery as well. Currently we are applying DbM in tandem with Bayesian optimization for the optimization of 3D airfoils and vertical-axis wind turbines.

Acknowledgment

The authors would like to thank Dr. Ömer Savaş, affiliated with the Department of Mechanical Engineering, University of California at Berkeley for insightful discussions regarding airfoils and aerodynamics. The authors would also like to acknowledge that this work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number ACI-1548562 through allocation TG-CTS190047.

Funding Data

No funding was received for the undertaking of this project.

Data and Materials Availability

The data needed to evaluate the conclusions are present in the paper and Appendices. The data files and optimization setup will be posted in a public repository upon publication of the paper.

Nomenclature

- $c =$ airfoil chord length (m)
- $d =$ drag force of an airfoil per unit span (N m$^{-1}$)
- $l =$ lift force of an airfoil per unit span (N m$^{-1}$)
- $P =$ $y$-coordinate collocation vector of a morphed airfoil
- $S =$ $y$-coordinate collocation vector of a baseline airfoil
Fig. 9 Mean weight distributions of the Pareto-optimal airfoil shapes with respect to twenty-five baseline airfoil shapes.

(a) Total mean

(b) PCA axis 1

(c) PCA axis 2

Fig. 10 Morphed airfoil shapes generated by the optimal weight vectors, representing (a) the total mean of all optimal airfoils’ weights, (b) the coefficients of the PCA axis having the most variance and (c) the coefficients of the PCA axis having the second-most variance. The black and red surfaces correspond to the first and second half of the collocation points, respectively.

Greek Letters

\[ \alpha = \text{airfoil angle of attack (°)} \]
\[ \alpha_s = \text{airfoil stall angle (°)} \]
\[ \Delta \alpha = \text{stall angle tolerance, the range of } \alpha \text{ between the stall point and the maximum lift-drag ratio point (°)} \]
\[ \nu = \text{fluid kinematic viscosity (m}^2\text{s}^{-1}) \]
\[ \rho = \text{fluid density (kg m}^{-3} \)]

Dimensionless Groups

\[ \frac{C_d}{\max \alpha} = \text{maximum lift-drag ratio of an airfoil, } \frac{C_l}{C_d} \]
\[ \frac{\text{Re}}{U_c/\nu} = \text{Reynolds number based on airfoil chord length, } \frac{U_c}{\nu} \]

\[ U = \text{free-stream flow speed (m s}^{-1}) \]
\[ w = \text{design-by-morphing weight factor} \]
Appendix A: Aerodynamic Optimization Objectives

Airfoil optimization has become common in aerodynamic design problems involving maximization of one or more performance parameters of an airfoils. We mainly consider the following 2 performance parameters: the lift-drag ratio and stall angle. Given the flow speed \( U \), fluid density \( \rho \) and airfoil chord length \( c \), the lift and drag coefficients of an airfoil per unit span at an angle of attack \( \alpha \), \( C_l \) and \( C_d \), are expressed as

\[
C_l(\alpha) \equiv \frac{l(\alpha)}{\frac{1}{2} \rho U^2 c}, \quad C_d(\alpha) \equiv \frac{d(\alpha)}{\frac{1}{2} \rho U^2 c}
\]

(A1)

where \( l \) and \( d \) are lift and drag force per unit span, respectively, both of which change with respect to \( \alpha \). In this paper, there parameters are predicted via XFOIL[44], an program for analysis of subsonic isolated 2D airfoils, with varying \( \alpha \) and then used for the optimization. Based on \( C_l \) and \( C_d \), the lift-drag ratio \( CLD \) is calculated as:

\[
CLD(\alpha) = \frac{C_l(\alpha)}{C_d(\alpha)}
\]

(A2)

On the other hand, we define the stall angle \( \alpha_s \) as an angle of attack where \( C_l \) reaches the first local maximum when we increase the angle starting from \( 0^\circ \), or

\[
\alpha_s \equiv \min_{\alpha \geq 0} \{ \alpha \mid \exists \delta > 0 \text{ such that } C_l(\alpha) \geq C_l(x) \forall x \in [\alpha - \delta, \alpha + \delta] \}
\]

(A3)

Note that this definition is more conservative than the typical definition of the stalling in practice, where flow at the rear region begins to fully separate and \( C_l \) is globally maximized. \( \alpha_s \) is occasionally smaller than the global maximum of \( C_l \). Nonetheless, this approach helps avoid overestimation of the stall angle, which is expected to happen in XFOIL, because of the nature of its flow solver having a limited accuracy in stall and post-stall conditions.

\( CLD \) and \( \alpha_s \) have been typically considered to be significant to characterize the airfoil performance. For example, when it comes to lift-type wind turbines, the point where \( CLD \) is maximized may be commonly chosen as the design point. Since a wind turbine cannot always operate in the design condition, however, \( \alpha_s \) needs to be additionally considered to evaluate how far the turbines run under an increasing-lift condition. For well-designed airfoils, \( \alpha_s \) generally occurs later than the design point, which yields tolerance in operation beyond the design point. Consequently, the stall angle tolerance, i.e. the range between these two angles of attack \( \Delta \alpha \), which is expressed as

\[
\Delta \alpha \equiv \max \left\{ 0, \alpha_s - \arg \max_{\alpha \in \mathbb{R}} CLD(\alpha) \right\}
\]

(A4)

can be a proper choice to evaluate the off-design performance [84]. Figure 11 depicts a scheme of how \( CLD \) and \( \Delta \alpha \) are determined on performance curves of an airfoil.

Appendix B: Baseline Airfoil Shapes and Validation

Our optimization methodology does not rely on one specific airfoil evaluation tool. To compare our results with the previous literature and to help future researchers quickly reproduce our results, we use XFOIL[44] in the present study. The two design objectives, \( CLD_{\text{max}} \) and \( \Delta \alpha \), are obtained from the \( C_l \) and \( C_d \) data calculated by the XFOIL at different angle of attacks (see Figure 11).

To achieve better efficiency and consistency, we only use the XFOIL to generate the performance data and do not use any of its built-in paneling features. The conditioning and the re-paneling of the morphed airfoil coordinates are handled at the end of our DbM algorithm. To reduce the evaluation time, we perform a rough scan first with an \( \alpha \) increment of \( 1^\circ \) and then finer scans for \( CLD_{\text{max}} \) and \( \Delta \alpha \) separately with an \( \alpha \) increment of 0.25\(^{\circ} \).

It is worth noting that XFOIL uses a global Newton’s method[44] to simultaneously solve for the boundary layer and the transition equations, and it uses the solution at the previous angle of attack as the starting guess. As a result, ill-conditioned airfoil coordinates and the occurrence of the flow separation can both lead to the non-convergence of the XFOIL evaluation. To ensure the robustness and correctness of our airfoil evaluation, our XFOIL wrapper attempts to reach convergence by re-starting the root-finding with a fresh starting guess and by gradually increasing the number of the panels. If both attempts fail, the wrapper will check the convergence at the neighboring points, which will indicate whether the flow separation occurs or not. Besides the non-convergence issue, we further verify the correctness of the Newton’s method by comparing the calculated viscous drag coefficient and the inviscid drag coefficient, the later of which is determined purely by the potential flow theorem and have to be smaller than its viscous counterpart due to its negligence of the friction (viscous effect). Any angle with incorrect result will undergo the same treatment as the non-converging ones, hence ensuring the correctness of our airfoil performance evaluation. A comparison between our XFOIL evaluation and the past experimental results of the same airfoil under the same flow conditions is provided in the table.

Appendix C: Optimization Test Functions and Validation

We use the multi-objective problems, suggested by Zeidtler et al.[82], for testing our GA setup. The details of the test functions are given in Table 2. All the test functions were minimized with 25 variables in the design space.

MATLAB’s NSGA-II genetic algorithm, a fast sorting and elitist multi-objective genetic algorithm, was used in the current study. Single objective optimization for each objective and random sampling were used for initialization. The population size of 372 was used with a total of 3,000 maximum generations. A ‘phenotype’ crowding distance metric was used. This setup was validated on the test functions described above. All the problems were benchmarked with 25 variables (\( d = 25 \)) and two objective functions (\( K = 2 \)) as with the present airfoil optimization problem. The results of our setup on four benchmark problems are shown in
Table 1  The model names, features, shape outlines, and XFOIL evaluation results of the 25 baseline shapes used by DbM in this paper. The coordinates of the baseline shapes are obtained from the UIUC airfoil coordinates database[40]. The airfoil evaluation results are obtained for an incompressible outer-flow of Re = 1 × 10^6. The reference evaluation results are interpolated from the Airfoil Tools online database[85], where N/A indicates that there is no data available for this airfoil.

<table>
<thead>
<tr>
<th>Index</th>
<th>Model Name</th>
<th>Series (Features)</th>
<th>Airfoil Shape</th>
<th>Reference[85]</th>
<th>Present</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CLD_max</td>
<td>Δα</td>
</tr>
<tr>
<td>1</td>
<td>NACA 0012</td>
<td>NACA (4-digit)</td>
<td></td>
<td>75.6</td>
<td>8.50</td>
</tr>
<tr>
<td>2</td>
<td>NACA 2412</td>
<td>NACA (4-digit)</td>
<td></td>
<td>101.4</td>
<td>12.00</td>
</tr>
<tr>
<td>3</td>
<td>NACA 4412</td>
<td>NACA (4-digit)</td>
<td></td>
<td>129.4</td>
<td>1.75</td>
</tr>
<tr>
<td>4</td>
<td>E 205</td>
<td>Eppler</td>
<td></td>
<td>128.3</td>
<td>8.50</td>
</tr>
<tr>
<td>5</td>
<td>AH 81-K-144 W-F Klappe</td>
<td>Althaus</td>
<td></td>
<td>89.7</td>
<td>2.00</td>
</tr>
<tr>
<td>6</td>
<td>AH 79-100 C</td>
<td>Althaus</td>
<td></td>
<td>183.0</td>
<td>14.75</td>
</tr>
<tr>
<td>7</td>
<td>AH 79-K-143/18</td>
<td>Althaus</td>
<td></td>
<td>110.9</td>
<td>1.50</td>
</tr>
<tr>
<td>8</td>
<td>AH 94-W-301</td>
<td>Althaus</td>
<td></td>
<td>103.0</td>
<td>4.00</td>
</tr>
<tr>
<td>9</td>
<td>NACA 23112</td>
<td>NACA (5-digit)</td>
<td></td>
<td>98.6</td>
<td>6.75</td>
</tr>
<tr>
<td>10</td>
<td>NACA 64(2)-415</td>
<td>NACA (6-digit)</td>
<td></td>
<td>120.6</td>
<td>12.50</td>
</tr>
<tr>
<td>11</td>
<td>NACA 747(A)-315</td>
<td>NACA (7-digit)</td>
<td></td>
<td>111.5</td>
<td>12.00</td>
</tr>
<tr>
<td>12</td>
<td>Griffith 30% Suction</td>
<td>Griffith (Suction)</td>
<td></td>
<td>17.3</td>
<td>0.00</td>
</tr>
<tr>
<td>13</td>
<td>AS 6097</td>
<td>Selig (Bird-like)</td>
<td></td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>14</td>
<td>E 379</td>
<td>Eppler (Bird-like)</td>
<td></td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>15</td>
<td>Clark YS</td>
<td>Clark</td>
<td></td>
<td>85.7</td>
<td>5.25</td>
</tr>
<tr>
<td>16</td>
<td>Clark W</td>
<td>Clark</td>
<td></td>
<td>116.1</td>
<td>11.00</td>
</tr>
<tr>
<td>17</td>
<td>Clark Y</td>
<td>Clark</td>
<td></td>
<td>114.8</td>
<td>11.75</td>
</tr>
<tr>
<td>18</td>
<td>Chen</td>
<td>Chen</td>
<td></td>
<td>125.4</td>
<td>0.00</td>
</tr>
<tr>
<td>19</td>
<td>S2027 Flipped</td>
<td>Selig (Flipped)</td>
<td></td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>20</td>
<td>GOE 417A</td>
<td>Gottingen (Thin plate)</td>
<td></td>
<td>86.7</td>
<td>5.25</td>
</tr>
<tr>
<td>21</td>
<td>GOE 611</td>
<td>Gottingen (Flat bottom)</td>
<td></td>
<td>125.6</td>
<td>9.00</td>
</tr>
<tr>
<td>22</td>
<td>Dragonfly Canard</td>
<td>Dragonfly</td>
<td></td>
<td>144.6</td>
<td>2.50</td>
</tr>
<tr>
<td>23</td>
<td>FX 79-W-470A</td>
<td>Wortmann (Fat)</td>
<td></td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>24</td>
<td>Sikorsky DBLN-526</td>
<td>Sikorsky (Fat)</td>
<td></td>
<td>53.3</td>
<td>4.75</td>
</tr>
<tr>
<td>25</td>
<td>FX 82-512</td>
<td>Wortmann</td>
<td></td>
<td>99.1</td>
<td>14.75</td>
</tr>
</tbody>
</table>

Figure 12. It was found that the algorithm could capture ZDT1, ZDT2, and ZDT4 accurately and predicts ZDT6, which is not only non-convex but also non-uniform, reasonably well.

Appendix D: Airfoil Shape Clustering

To analyze characteristics of the optimized airfoil shapes in detail, the airfoil shapes on the Pareto front are classified into 3 clusters using k-means clustering based on the Euclidean distance with k = 3. The clustering is performed in the design variable space, or weight space, of D^25 rather than in the objective plane because the purpose of clustering is to identify common geometric features over different airfoil shapes as a result of the optimization. The selection of the cluster size is based on the PCA of the optimal weight vector set.

Figure 13 shows the projection of the 25-dimensional weight vector set to the 2-dimensional subspace spanned by 2 PCA axes having the first- and second-most variance. The explained variance ratios of PCA axes 1 and 2 are 77.8% and 14.6%, respectively. On the other hand, the PCA axis of the third-most variance accounts for only 2.5% of the variance, affirming that the 2-dimensional projection in Figure 13 adequately scatters the clusters. From this observation, k = 3 is thought to be the most appropriate cluster size.
Table 2  Benchmark Test Functions. All of the test functions are bi-objective with extended to $n$-dimensional constrained search space.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Bounds</th>
<th>Objective Functions</th>
<th>Optima</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ZDT_1$</td>
<td>$w_i \in [0, 1], \ i = 1, \ldots, n$</td>
<td>$f_1(w) = w_1$</td>
<td>$w_1 \in [0, 1]$</td>
<td>convex</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$f_2(w) = g(w) \left[ 1 - \left( \frac{f_1(w)}{g(w)} \right)^{1/2} \right]$</td>
<td>$w_i = 0, \ i = 2, \ldots, n$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$g(w) = 1 + 9 \left( \sum_{i=2}^{n} w_i \right) / (n - 1)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$ZDT_2$</td>
<td>$w_i \in [0, 1], \ i = 1, \ldots, n$</td>
<td>$f_1(w) = w_1$</td>
<td>$w_1 \in [0, 1]$</td>
<td>non-convex</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$f_2(w) = g(w) \left[ 1 - \left( \frac{f_1(w)}{g(w)} \right)^2 \right]$</td>
<td>$w_i = 0, \ i = 2, \ldots, n$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$g(w) = 1 + 9 \left( \sum_{i=2}^{n} w_i \right) / (n - 1)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$ZDT_4$</td>
<td>$w_1 \in [0, 1]$</td>
<td>$f_1(w) = \exp(-4w_1) \sin^6(6\pi w_1)$</td>
<td>$w_1 \in [0, 1]$</td>
<td>non-convex</td>
</tr>
<tr>
<td></td>
<td>$w_i \in [-5, 5], \ i = 2, \ldots, n$</td>
<td>$f_2(w) = g(w) \left[ 1 - \left( \frac{f_1(w)}{g(w)} \right)^2 \right]$</td>
<td>$w_i = 0, \ i = 2, \ldots, n$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$g(w) = 10n + \sum_{i=2}^{n} (w_i^2 - 10\cos(4\pi w_i)) - 9$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$ZDT_6$</td>
<td>$w_i \in [0, 1], \ i = 1, \ldots, n$</td>
<td>$f_1(w) = w_1$</td>
<td>$w_1 \in [0, 1]$</td>
<td>non-convex, non-uniform</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$f_2(w) = g(w) \left[ 1 - \left( \frac{f_1(w)}{g(w)} \right)^2 \right]$</td>
<td>$w_i = 0, \ i = 2, \ldots, n$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$g(w) = 1 + 9 \left( \sum_{i=2}^{n} w_i \right) / (n - 1)^{1/4}$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 12  Multi-objective optimization of benchmark test functions using GA

Fig. 13  Projection of the 25-dimensional optimal weight vectors to the 2-dimensional subspace spanned by 2 PCA axes of the dominant variance. $k$-means clustering with the cluster size of 3 is used to identify the clusters.


