Airfoil Shape Optimization: Comparative Study of Meta-heuristic Algorithms, Airfoil Parameterization Methods and Reynolds Number Impact

José D Hoyos, Jesús H Jiménez, Camilo Echavarría, Juan P Alvarado
Aeronautical Engineering Faculty, University Pontificia Bolivariana, Medellín, Colombia
E-mail: jdaniel.hoyos@udea.edu.co

Abstract. The aerodynamic efficiency in airfoil theory is defined as the ratio between the lift and drag force, which is the main objective function to be maximized in a wide kind of vehicle design due to its strong relationship between fuel consumption and range. This work employs the 4-digits NACA parameterization, a recently developed 6-parameters method, and the PARSEC technique with a correction of the matrices available in the literature, to compare the computational cost and the ability to achieved higher efficiency of these parameterizations. A genetic algorithm and particle swarm optimization routines are developed and implemented in Matlab, also a sine-cosine algorithm is tested, where Xfoil and the open-source computational fluid dynamic software OpenFOAM are coupled with the optimization algorithms. Finally, a Reynolds number impact study is performed related to the airfoil shape and the angle of attack which maximizes the aerodynamic efficiency. The results showed a faster convergence for the particle swarm optimization and the highest aerodynamic efficiency achieved by the 6-parameter method. Furthermore, with a higher Reynolds number, a higher angle of attack for the optimum lift-to-drag ratio as well a less camber is obtained.

1. Introduction
Nowadays one of the major concerns on airfoil implementation is to assure suitable aerodynamic characteristics to a defined operating condition, the lift-to-drag ratio is commonly one of the most dominant indicators which are highly correlated and affects directly aircraft performance. So far, there are several already designed airfoils that can be adopted, however those can present an efficiency peak on a particular and different application. Airfoil parameterization methods appear to present a novel formulation of both camber and thickness line in terms of several parameters where an original aerodynamic surface can be developed. In this sense, control and management of such variables must be applied in order to avoid useless airfoil generation, this can be translated establishing conscientiously limits in the parameters of the formulation. On the other hand, an optimization routine could be presented as an implementation attempting to search and reach an optimum value associated with a defined variable during any process with a set of constraints. The previously mentioned allows to establishment a convenience to link these two implementations where the employment of a computational tool to extract the aerodynamic coefficients will be required in order to find finally an optimum new airfoil shape for a defined operating condition.
This work aims to compare some conventional and novel methods of airfoil parameterization coupled with different meta-heuristic optimization algorithms, that is, algorithms that work like a black box, where a possible constraint optimization problem enters and an optimum is obtained, the nature of the problem does not affect the operation of the meta-heuristic method. The objective function to optimize is the lift-to-drag ratio, also called aerodynamic efficiency, at the certain wind and angle conditions. The search space is the airfoil shape.

The parameterization methods are 4-digit NACA [1] which has three parameters, another with six parameters [2] and the PARSEC method [3, 4] which employ eleven parameters. The ranges for the parameters are found and some corrections to the PARSEC method are proposed. The tested algorithms are the Particle Swarm Optimization (PSO) [5], the Genetic Algorithm (GA) [6] and the Sine-Cosine Algorithm (SCA) [7].

Moreover, a comparison between the optimum airfoil shapes achieved by Xfoil and OpenFOAM is carried out, where mesh independence and turbulence study is performed. The automatic routines are implemented in Matlab coupled with Xfoil and OpenFOAM. Finally, the Reynolds (Re) number impact in the airfoil shape is studied with the NACA parameterization coupled with Xfoil and PSO also varying the angle of attack.

2. Airfoil parameterization

2.1. Three parameters method

The airfoil shape generator method with three parameters selected is the 4-digit NACA family [1]. The shape is described using a series of four digits that are entered into equations to precisely generate the shape. The first digit specifies the maximum camber \( m \) in chord percentage, the second indicates the position of the maximum camber \( p \) in tenths of the chord, and the last two numbers provide the maximum thickness \( t \) of the airfoil in chord percentage. The thickness distribution for a closed trailing edge is shown in equation 1.

\[
\pm y_t = \frac{t}{0.2} \left(0.2968\sqrt{x} - 0.126x - 0.3516x^2 + 0.2843x^3 - 0.1036x^4 \right)
\] (1)

The mean camber line \( y_c \) coordinate is defined in two equations (2 and 3) as function of \( x \) coordinate.

\[
y_c = \frac{m}{p^2} (2px - x^2) \quad \text{for} \quad 0 \leq x \leq p
\] (2)

\[
y_c = \frac{m}{(1-p)^2} (1 - 2p + 2pc - x^2) \quad \text{for} \quad p < x \leq c
\] (3)

Finally, the coordinates of upper surface \((x_u, y_u)\) and lower surface \((x_l, y_l)\) are calculated from equation 4 to equation 7

\[
x_u = x - y_t \sin(\theta)
\] (4)

\[
x_l = x + y_t \sin(\theta)
\] (5)

\[
y_u = y_c + y_t \cos(\theta)
\] (6)

\[
y_l = y_c - y_t \cos(\theta)
\] (7)

where \( \theta = \arctan(\frac{dy_c}{dx}) \).

Some combinations of the parameters generate intersecting geometries or very thin or thick shapes which Xfoil cannot solve, so the following ranges are established for the variables: \( m \in [0, 0.12], p \in [0.2, 0.7] \) and \( t \in [0.07, 0.25] \).
2.2. Six parameters method
A recent parametric model for airfoil shape with six parameters is presented in [2]. The model has some interesting characteristics, the parameters have an intuitive meaning, and their change predictably impacts the airfoil. The generated curve is continuous and smooth, reducing the design space to realistic shapes. The $x$ and $y$ coordinates are described by equations from 8 to 10.

\[
y(\theta) = \frac{T}{2} \left| \sin \theta \right|^B (1 - x^P) + C \sin(x^E \pi) + R \sin(x^{2\pi}) \tag{8}
\]
\[
x(\theta) = 0.5 + 0.5 \left| \cos \theta \right|^B \cos \theta \tag{9}
\]
\[
\theta \in [0, 2\pi] \tag{10}
\]

Where $B$ is the base shape coefficient. For $B=2$ the base shape is an ellipse. $T$ is the thickness as a chord fraction. $P$ is the taper parameter. $C$ is the camber as a fraction of the chord. $E$ is the camber parameter. $R$ is a reflex parameter, a positive value generates a reflexed trailing edge. An example of the wide range of shapes that can be generated by this method is shown in figure 1.

![Figure 1: Airfoil for $B=2$, $T=0.1$, $C=0.05$, $P=1$, $E=1$, and $R=0$, where one of the parameters is changed, from [2].](image)

The selected ranges for the optimization meet the same thickness and camber restrictions as the NACA parameterization: $B \in [1.95, 2.3]$, $T \in [0.085, 0.2]$, $P \in [0.85, 1.5]$, $C \in [0, 0.12]$, $E \in [0.65, 1.4]$ and $R \in [-0.015, 0.015]$. Some of the limits are imposed because Xfoil has problems working outside these values.

2.3. Eleven parameters method
The method selected is the PARSEC parameterization [3]. This parameterization has previously been coupled with genetic algorithms [4]. It uses eleven basic parameters to completely define the airfoil shape as shown in figure 2.

![Figure 2: Airfoil shape defined by eleven PARSEC parameters.](image)
$Z_{X_{up}}$ and $Z_{X_{low}}$ are the upper and lower values of curvature.

The parameter $\Delta Z_{te}$ is fixed, hence the parameterization is summarized to 10 variables. In this method, a linear combination of shape functions describes the aerofoil shape as 11 shows.

$$z_{up,lo} = \sum_{i=1}^{6} a_{up,low}^i \cdot x^{i-\frac{1}{2}}$$ (11)

$$a_{up} = C_{up}^{-1} b_{up}$$ (12) $$a_{low} = -C_{low}^{-1} b_{low}$$ (13)

$$C_{up,low} = \begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 \\
X_{up,low}^{\frac{1}{2}} & X_{up,low}^{\frac{3}{2}} & X_{up,low}^{\frac{5}{2}} & X_{up,low}^{\frac{7}{2}} & X_{up,low}^{\frac{9}{2}} & X_{up,low}^{\frac{11}{2}} \\
\frac{1}{2} & 3 & 5 & 7 & 9 & 11 \\
\frac{1}{2} X_{up,low}^{\frac{1}{4}} & 3 X_{up,low}^{\frac{3}{4}} & 5 X_{up,low}^{\frac{5}{4}} & 7 X_{up,low}^{\frac{7}{4}} & 9 X_{up,low}^{\frac{9}{4}} & 11 X_{up,low}^{\frac{11}{4}} \\
-\frac{1}{4} X_{up,low}^{\frac{1}{8}} & 3 X_{up,low}^{\frac{3}{8}} & 15 X_{up,low}^{\frac{5}{8}} & 15 X_{up,low}^{\frac{7}{8}} & 63 X_{up,low}^{\frac{9}{8}} & 99 X_{up,low}^{\frac{11}{8}} \\
1 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

$$b_{up} = \begin{bmatrix}
\frac{\Delta Z_{te}}{2} + Z_{te} & Z_{up} & \tan\left(-\frac{\beta_{te}}{2} - \alpha_{te}\right) & 0 & Z_{X_{up}} & \sqrt{2r_{te}}
\end{bmatrix}^T$$

$$b_{low} = \begin{bmatrix}
\frac{\Delta Z_{te}}{2} - Z_{te} & Z_{low} & \tan\left(\frac{\beta_{te}}{2} + \alpha_{te}\right) & 0 & Z_{X_{low}} & \sqrt{2r_{te}}
\end{bmatrix}^T$$

A slight variation for this method is proposed. This parameterization lacks on camber control, besides, the parameter $Z_{te}$ moves up or down the trailing edge, tilting the chord line. The airfoil shape is rotated after generated in such a way that the trailing edge is located exactly at the horizontal axis, giving a little more control of the airfoil curvature and making this parameterization some wider.

The following ranges are selected to generate the airfoil shapes: $r_{te} \in [0.008, 0.025]$, $X_{up} \in [0.317, 0.45]$, $Z_{up} \in [0.03, 0.12]$, $Z_{X_{up}} \in [-2, 0]$, $X_{low} \in [0.3, 0.45]$, $Z_{low} \in [0.01, 0.06]$, $Z_{X_{low}} \in [-2.1]$, $Z_{te} \in [-0.25, 0]$, $\Delta Z_{te} = 0.006$, $\alpha_{te} \in [0, 0.436]$ and $\beta_{te} \in [0, 0.35]$.

3. Polar airfoil data
The aerodynamic coefficients are computed through the potential flow based software Xfoil and finite volume software OpenFOAM to test different models formulation where computational cost and extracted aerodynamic data could differ, resulting in a slight variation of the optimum airfoil shape.

3.1. Xfoil
Xfoil is a computational implementation of a panels method that incorporates a $e^n$ laminar to turbulent transition technique coupled with an integral boundary layer formulation [8], which allows to predict $C_l$ and $C_d$ at small angles of attack. A Matlab code is developed to extract
aerodynamic data for each mutated airfoil set by the optimization algorithm combined with the airfoil parameterization. A recursion within the code is implemented increasing the airfoil surface nodes as the maximum number of iteration, attempting to guarantee convergence of the solution avoiding random Xfoil issues during the calculation process. The airfoil is modeled with 250 points over the top and bottom surfaces applying a smoothing step fixing any high variation of the coordinates, the calculation is set with 300 available iteration and the viscous mode is activated to compute correctly the drag coefficient.

3.2. CFD: OpenFOAM

OpenFOAM is an open-source CFD software under the GNU General Public License which solves the main fluid governing equations through the finite volume method within the discretized computational domain [9], with the possibility to include additional fluid variables associated to already defined turbulence model. A Matlab code is developed to connects with OpenFOAM and compute automatically the aerodynamic coefficients for each airfoil shape established.

3.2.1. Mesh topology

The Mesh is generated from the open-source software construct2d [10], which implements an elliptical or hyperbolic algorithm to develop a completely structured either C-Grid or O-grid topology, commonly used for airfoil analysis on CFD simulation. This tool allows us to interact through the operating system terminal command line, allowing us to link a Matlab script to generate automatically the mesh for any airfoil.

The O-grid topology is selected to guarantee a suitable refinement in any flow direction with a domain diameter of $25c$ with $c$ being the chord length [11, 12]. The elliptical mesh algorithm is selected which perform additionally local smoothing steps near the airfoil wall, obtaining a higher cells refinement near the airfoil surface where the fluid variables gradients are more notables, accomplishing well transfer fluid variables information between the cells, this results on less number of cells closer the domain boundaries where fluid properties present small changes. In terms of mesh quality a $y^+ < 1$ is guaranteed [13] employing the flat-plate boundary layer theory [14, p. 467], where $y^+$ is the non-dimensional wall distance, finally the average cell skewness angle is less than 0 and the average cell aspect ratio is under 30.

3.2.2. Simulation Setup

The Matlab code coupled to OpenFOAM allows us to establish any desired setup, varying the main simulation parameters like convergence criteria, numerical schemes, turbulence model, etc. For this particular case, the simulation is performed with a steady-state time scheme and Gauss linear divergence scheme [15] at standard sea level condition, the SIMPLE algorithm is implemented and the convergence criteria are set on $1 \times 10^{-4}$ for the residuals of the fluid variables. The upstream domain side is defined as a velocity inlet boundary condition, the downstream domain side is set as a pressure outlet boundary condition with gauge pressure equal to zero and the airfoil is set as a wall boundary with no-slip condition. The angle of attack is guarantee setting the respective wind speed $V_x$ and $V_y$ components in order to use one single mesh for any angle. The normal and axial aerodynamic forces over the airfoil are computed, hence, must be transformed into the relative wind reference frame and then be normalized to obtain the lift and drag coefficient.

3.2.3. Grid independence and turbulence model

To establish a number of cells that assure accurate data with the lowest computational time, a grid independence studied is performed, varying the number of grid cells between a range of $1.5 \times 10^4$ and $15 \times 10^4$ volumetric elements. The NACA 0015 airfoil is selected which has reported wind tunnel test data by Sheldahl and Klimas [16]. The Re number is selected as $1 \times 10^6$ with an angle of attack $\alpha = 0^\circ$, results are shown in figure 3.
To select the most suitable turbulence model three options are considered. The Spalart-Allmaras which implements only one turbulence equation but is often used for low Re number applications [17], the $\kappa-\omega$ SST and a four-equation model Langtry-Menter $\kappa-\omega$ SST which adds two other equations: one for the intermittency $\gamma$ and other for the laminar-turbulent transition with $Re_{\theta_t}$ criteria, which links empirical transition data with intermittency equation [18].

![Grid independence and computational time analysis for O-Grid topology, lift coefficient (a) and drag coefficient (b).](image)

Figure 3: Grid independence and computational time analysis for O-Grid topology, lift coefficient (a) and drag coefficient (b).

![O-grid topology generated by Construct2d with $5 \times 10^4$ cells.](image)

Figure 4: O-grid topology generated by Construct2d with $5 \times 10^4$ cells.

It is shown from the results that $C_l$ and $C_d$ presents a good match in comparison with the
experimental data. The $C_l$ is close enough to zero in term of magnitude order from $5 \times 10^4$ number of cells for all turbulence models implemented, the $C_d$ show a better fit employing the Langtry-Menter-$\kappa-\omega$ SST, also termed as the $\gamma-Re_\theta$-SST model [19], with a minimum relative error of 11% at a $5 \times 10^4$ number of cells, value where the mesh convergence is reached, so this number of cells is employed in the optimization.

### 3.3. Validation

Aerodynamic coefficients are computed for the NACA 0015 airfoil from $\alpha = 0^\circ$ to $\alpha = 15^\circ$ at a $Re$ number of $1 \times 10^6$ through Xfoil and CFD OpenFOAM and are compared against experimental data, the results are shown in figure 5.

![Figure 5: Wind tunnel data, CFD OpenFOAM and Xfoil Lift coefficients (a) and Drag coefficients (b) validation for a $Re = 1 \times 10^5$.](image-url)

From the results obtained, the lift coefficient extracted from CFD OpenFOAM and Xfoil present an average error of 8.21% and 8.23% respectively, highlighting the stall prediction by the finite volume software which is an important indicator on aerodynamics coefficient calculation, on the other hand, the average error associated to $C_d$ is 8.09% and 6.78% for CFD OpenFOAM and Xfoil respectively, resulting on a quite good match in comparison with the wind tunnel data for both tools in this $Re$ number range.

### 4. Meta-heuristic algorithms

#### 4.1. Genetic algorithm

The GA method mimics the natural evolution, where some initial population is selected, then a crossover and mutation operations are performed usually based on the score of each individual by the objective function.

##### 4.1.1. Initial population

The initial population has a certain number of individuals and for each individual, the values for the variables are randomly distributed in the design space taking into account the maximum and minimum limits for each design variable.
4.1.2. Selection of parents

The selection operator must choose with higher probability those individuals who present a better value of the fitness function, also including the possibility that individuals who are not as good have a certain probability of appearing. The latter tries to keep a certain diversity in regarding the information that the chromosomes contain. The objective is that the quantity of each one of the individuals in the new population is given by the proportion given in equation 14.

\[
P(x_i) = \frac{f(x_i)}{\sum_{j=1}^{N_{ind}} f(x_j)}
\]  

(14)

where \( N_{ind} \) is the number of individuals, \( P(x_i) \) is the probability of the selection for the \( x_i \) individual and \( f(x_i) \) is the value of the function to be optimized. In this work, the Stochastic Universal Sampling (SUS) is employed to select the parents, which fulfill the proportion by equation 14. In order to avoid convergence in local optimums, a linear ranking function is employed, in this method, the individuals are ordered from highest to lowest in an array according to the fitness function value, and their position in the array is the new value of the fitness function. An example of the SUS methodology with and without ranking function is shown in figure 6.

![Stochastic Universal Sampling 8 parents selection from 6 individuals without (a) and with ranking function (b).](image)

Figure 6: Stochastic Universal Sampling 8 parents selection from 6 individuals without (a) and with ranking function (b).

4.1.3. Cross over and mutation

Cross over is performed to combine the desirable characters of two different parents selected by the SUS-linear ranking method. In this work, uniform cross over is employed. In this approach, a cross over probability \( P_c \) is defined and a probability test is performed for each chromosome. Since the coding of the problem is not binary, it is possible to perform the linear recombination to the chromosomes of the real variable to increase the diversity of the individuals shown in equations 15 and 16.

\[
C_1 = \alpha_1 P_1 + (1 - \alpha_1) P_2
\]  

\[
C_2 = \alpha_2 P_2 + (1 - \alpha_2) P_1
\]  

(15)  \hspace{1cm} (16)

where \( C_n \) is the child \( n \), \( P_n \) is the parent \( n \) and \( \alpha_1, \alpha_2 \) are random numbers between \(-0.5\) and \(0.5\), which extends the linear combination beyond the limits set by the parents.

The mutation is performed to increase the diversity of genes and thus have a greater probability of success by finding global optima. Here a mutation probability \( P_m \) is defined and the probability test is carried out on each chromosome. If passed, the mutation is performed adding a low random number relative to chromosome value.
4.1.4. Parameters setting

The main parameters to adjust are the mutation probability \( P_m \) and the cross over probability \( P_c \). Due to the unknown nature of the function, parameters are chosen that have shown a good result for all types of function tested in [6], \( P_m = 0.1, P_c = 0.6 \). Moreover, the number of individuals in one generation \( N_{ind} \) is chosen as \( 10 \cdot n \) and the maximum number of generations as \( 25 \cdot n \) where \( n \) is the number of variables.

4.2. Particle swarm algorithm

The particle swarm optimization (PSO) method attempts to find an optimum design by moving a given number of possible solutions (the particles) along the design space of the optimization problem, which can be subjected to some constraints.

The movement of the particles is not completely random. The positions taken by particles are evaluated in order to define their further movement; the trajectory and velocity of an specific particle depend on the best position that has reached any particle at any time. A simple variant for the method establishes that the trajectory of a specific particle is affected also by its individual best position so local optimal values are avoided. Equations 17 and 18 are used to define the movement of an individual particle at a given time step \( (t) \) [5].

\[
X_{(t+1)} = \mu X_{(t)} + V_{(t+1)} \quad (17)
\]

\[
V_{(t+1)} = \mu V_{(t)} + \phi_1 R_2 \otimes (G_{(t)} - X_{(t)}) + \phi_2 R_2 \otimes (L_{(t)} - X_{(t)}) \quad (18)
\]

In this case, the operator \( \otimes \) denotes the array element-by-element product. The design parameters are stored at the array \( X \), the \( V \) array represents how the particle will be relocated at the next time step. \( G \) stores the best design reached by any particle while \( L \) represents the best design the individual particle reached. Look that speed depends on how far the particle is from the best one as well as from the local best, constants \( \phi_1 \) and \( \phi_2 \) are used to manipulate how strong the particle is attracted by the best one or by its individual best position. \( R_1 \) and \( R_2 \) are random vectors with many elements as many design variables the problem has. These random values vary between 0 and 1 giving randomness to the movement of the particles in order to improve the search. Eventually, after some time steps (or iterations) most of the particles will converge at an optimal design. The algorithm can be stopped by setting convergence criteria or well a maximum number of iterations. For this case the particles are randomly located on the design space; when a particle goes out of the boundary, a penalty function is applied so it will not attract the other particles to an undesired position.

4.2.1. Parameters setting

A fixed inertia weight of \( \mu = 0.5 \) is used as a variant for the method to restrict a little the particle’s movement so they do not move unsteadily within the design space; \( \phi_1 \) and \( \phi_2 \) are both fixed to 2, meaning that local and global bests can have the same impact over a particle’s trajectory. The number of particles is selected to be \( 10 \cdot n \) and the maximum number of iterations is set as \( 25 \cdot n \).

4.3. Sine cosine algorithm

An interesting optimization method called Sine Cosine algorithm (SCA) is chosen for the analysis due to the fact that it is an algorithm that easily allows limits to be used for the design variables, and it was previously employed for geometric airfoil optimization [7].

The SCA creates multiple initial random search agents (solutions) and requires them to oscillate outwards or towards the best candidate employing a sine and cosine functions. The algorithm employs several random and adaptive variables integrated to emphasize exploration and exploitation of the search space, however, this method is employed as is discussed in [7],

\[
X_{(t+1)} = \mu X_{(t)} + V_{(t+1)} \quad (17)
\]

\[
V_{(t+1)} = \mu V_{(t)} + \phi_1 R_2 \otimes (G_{(t)} - X_{(t)}) + \phi_2 R_2 \otimes (L_{(t)} - X_{(t)}) \quad (18)
\]
no emphasis is placed on details. The number of search agents and the maximum number of iterations are also select as $10 \cdot n$ and $25 \cdot n$, respectively.

5. Results
The optimizations algorithms are compared for each parameterization method, the objective function is the aerodynamic efficiency $L/D$ for a Reynolds number of Re=1e6 and 5° angle of attack. The results show a higher optimum achieved by the GA and PSO algorithms compared to the SCA for all parameterizations as is shown in figures 7a, 7b and 8a. Moreover, the SCA shows a staggered search for the optimum, while PSO and GA quickly and smoothly reach the global optimum.

Although the PSO and GA achieve virtually the same optimum, the PSO achieves the optimum in a slightly smaller number of iterations as shown in figures 7a and 8a. Furthermore, the PSO does not evaluate the particles generated outside the limits of the variables, so it performs fewer evaluations of the objective function for the same number of iterations, considerably reducing the computation time as evidenced in the data in figures from 7a to 8a. If the number of evaluations of the objective function is compared, the rapid convergence of the PSO with respect to the GA is more marked.

The best airfoil shape from NACA parameterization is described by the values $t = 0.1000$, $m = 0.0992$ and $p = 0.5475$. From the limits imposed on these variables, it is observed that no parameter reached the limit value of the variable.
Similarly, the six parameters method achieved the optimum with the values \( B = 1.9501, T = 0.0850, P = 0.8500, C = 0.1047, E = 0.7821 \) and \( R = -0.015 \), taking into account the limits set for the parameters, it is observed that \( B, T, P \) and \( R \) takes the lower limits. According to these results, an airfoil with a lower thickness would achieve more efficiency, however, the optimization obtained limit values, which means that Xfoil limits the design for this 6-number parameterization.

The PARSEC method achieved an optimum with the following values \( r_{le} = 0.0169, X_{up} = 0.4108, Z_{up} = 0.0973, Z_{Xup} = -0.0101, X_{low} = 0.3937, Z_{low} = 0.0180, Z_{Xlow} = -1.3071, Z_{te} = -0.0783, \Delta Z_{te} = 0.006, \alpha_{te} = 0.3588 \) and \( \beta_{te} = 0.0025 \). From this values, any parameter reached the limit values.

The optimal thickness according to NACA parameterization is higher compared to the parameterization of 6 values, however both methods reach an optimum for the same camber value of 10%.

The norm of the subtraction of the vectors that contain the optimal parameters is employed to calculate the distance between the solutions found by each optimization algorithm, the results are shown in table 1. From the results, it is found that for each airfoil parameterization method, the lower difference is found comparing the GA and PSO algorithms.

| Method            | PSO-SCA | GA-PSO | GA-SCA |
|-------------------|---------|--------|--------|
| NACA-4 Digits     | 0.0148  | 0.0008 | 0.0152 |
| 6 Parameters      | 0.0416  | 0.0186 | 0.0432 |
| PARSEC            | 1.6311  | 0.3070 | 1.3428 |

Table 1: Solution vectors distance.

Finally, the PSO algorithm coupled with 4-digit NACA parameterization is employed to optimize the airfoil through OpenFOAM CFD code. The optimum parameters was \( t = 0.0787, m = 0.1055 \) and \( p = 0.3556 \), as it shown in figure 9a, less thickness than Xfoil method, with the maximum camber position closer to the leading edge, but about the same maximum camber of 10%. This airfoil is simulated at various angles of attack to compare Xfoil and OpenFOAM efficiency curves. It is shown in figure 9b that the airfoil was designed unintentionally by the PSO with the best aerodynamic efficiency at the arbitrarily selected angle for the optimization.

![NACA 4 Digits - PSO](image1)

Figure 9: Optimum airfoil comparison between Xfoil and OpenFOAM (a) and efficiency polar for OpenFOAM optimum airfoil comparison between Xfoil and OpenFOAM (b).

A 4-digits NACA airfoil optimization process varying the angles of attack with the constraints \( m \in [0, 0.12], p \in [0.25, 0.65] \) and \( t \in [0.08, 0.2] \) was carried out for different Re numbers. The optimum characteristics are shown in table 2 and the optimum shape for the lowest and highest
Re number are shown in figure 10. The camber has an increasing trend with the Re number, meanwhile the best efficiency angle of attack has the opposite behavior. Finally, the thickness achieved the lower limit from the constraint, which is the opposite behavior when optimizing by setting an angle of attack.

Re number | 2e4 | 5e4 | 1e5 | 2e5 | 4e5 | 1e6
---|---|---|---|---|---|---
Thickness | 0.0820 | 0.0800 | 0.0800 | 0.0807 | 0.0800 | 0.0802
Maximum camber | 0.0385 | 0.0444 | 0.0654 | 0.0609 | 0.0729 | 0.0808
Maximum camber position | 0.5720 | 0.4422 | 0.4570 | 0.5340 | 0.5669 | 0.5618
\( \alpha \) best efficiency | 7° | 7° | 7° | 5° | 4° | 3°
Aerodynamic efficiency | 22.4 | 43.3 | 68.5 | 203.1 | 148.9 | 222.2

Table 2: Optimum airfoil characteristics for different Re numbers.

![NACA 4 Digits - PSO - Xfoil](image)

Figure 10: Re number impact in airfoil shape optimization.

6. Conclusions
The GA and PSO algorithms reached almost the same optimum in all tested parameterizations, while the SCA reached a lower optimum for the same cases. However, the PSO algorithm achieved a convergence in fewer iterations in NACA parameterization, even achieving a considerably lower computation time for all parameterizations.

The 4-digit NACA parameterization reaches an optimum 17% lower in comparison to the parameterization of 6 values, however, it requires approximately 3.7 times less due to the fact that algorithms were run with populations and iterations depending on the number of variables. According to this, whether it is desired to achieve maximum efficiency regardless of the computational cost, the 6-value parameterization is suitable, otherwise the 4-digit NACA method achieves enough efficiency in a short time.

The NACA optimization obtained an optimum inside the limits, while the optimization of 6 values obtained four limit values, which were imposed to avoid errors in the convergence and reading of the airfoil by Xfoil. It is concluded that a more optimal airfoil can be achieved by the parameterization of 6 values using another tool to compute the aerodynamic coefficients such as CFD tools, while the NACA parameterization has an adequate match with Xfoil capabilities.

A remarkable result is that both the NACA and the 6-value parameterization, including Xfoil and OpenFOAM methods, obtained an optimal camber of around 10%, while the thickness did not remain constant in the solutions.

The PARSEC method reached an intermediate aerodynamic efficiency for the longest computation time, so this method is not recommended for optimization from a random population.
The optimization was carried out by setting an angle of attack without a guarantee that the airfoil presents its peak efficiency at this angle, however, the results show that the airfoils generated have the maximum efficiency at the design value. Noticeably, this does not mean that a higher optimum cannot be achieved if the angle of attack is added as an optimization parameter, which actually happened in the Re number study.

The optimum airfoil shape shows an inverse behavior between the Re number and the angle of attack at which the best aerodynamic efficiency is achieved, concurrent, the camber increases with Re number.

References

[1] Eastman N Jacobs K E W and Pinkerton R M 1935 The characteristics of 78 related airfoil sections from tests in the variable-density wind tunnel Tech. rep. Washington, D.C.

[2] Ziemkiewicz D 2017 *AIAA Journal* **55** 4390–4393 (Preprint https://doi.org/10.2514/1.J055986) URL https://doi.org/10.2514/1.J055986

[3] Sobieczky H 1997 *Geometry Generator for CFD and Applied Aerodynamics* (Vienna: Springer Vienna) pp 137–157 ISBN 978-3-7091-2658-5 URL https://doi.org/10.1007/978-3-7091-2658-5_9

[4] Della Vecchia P, Daniele E and D Amato E 2014 *Aerospace Science and Technology* **32** 103 – 110 ISSN 1270-9638 URL http://www.sciencedirect.com/science/article/pii/S1270963813002046

[5] Parsopoulos K E and Vrahatis M N 2010

[6] Ferragud F X B 1999 *Control predictivo basado en modelos mediante técnica de optimización heurística. Aplicación a procesos no lineales y multivariables* Ph.D. thesis URL http://hdl.handle.net/10251/15995

[7] Mirjalili S 2016 *Knowledge-Based Systems* **96** 120 – 133 ISSN 0950-7051 URL http://www.sciencedirect.com/science/article/pii/S0950705115005043

[8] Drela M 1989 *Low Reynolds number aerodynamics* (Springer) pp 1–12

[9] Open C 2011 *OpenFOAM Foundation* 2

[10] 2012 Open source software construct2d URL https://sourceforge.net/projects/construct2d

[11] Morgado J, Vizinho R, Silvestre M and Páscoa J 2016 *Aerospace Science and Technology* **52** 207–214 ISSN 1270-9638

[12] Günel O, Koç E and Yavuz T 2016 2016 *IEEE International Conference on Renewable Energy Research and Applications (ICRERA)* pp 628–632

[13] Salim S and Cheah S 2009 Proceedings of the International MultiConference on Engineering and Computer Science, Hong Kong

[14] White F 2003 *Fluid Mechanics* 5th ed McGraw-Hill series in mechanical engineering (New York: McGraw Hill)

[15] Suvanjumrat C 2017 *Engineering Journal* **21** 207–221

[16] Sheldahl R E and Klimas P C 1981 Aerodynamic characteristics of seven symmetrical airfoil sections through 180-degree angle of attack for use in aerodynamic analysis of vertical axis wind turbines Tech. rep. Sandia National Labs., Albuquerque, NM (USA)

[17] Islam M, Amin M R and Shariff Y M 2009 *ASME International Mechanical Engineering Congress and Exposition* vol 43826 pp 867–874

[18] Wang S, Ingham D B, Ma L, Pourkashanian M and Tao Z 2012 *Journal of Fluids and Structures* **33** 191 – 209 ISSN 0889-9746

[19] Menter F R, Langtry R and Völker S 2006 *Flow, Turbulence and Combustion* **77** 277–303