The Application of Evolutionary Computation in the Design of Wing Shapes

Leonid Scott
Division of Science and Math
University of Minnesota, Morris
Morris, Minnesota 56267
scot0530@morris.umn.edu
github.com/morrislenny

Abstract

A foil is a wing-like surface that generates lift and drag when moving through a fluid such as air or water. Foils can be found anywhere from the wings of a Boeing 747, to container ship propellers, sails, and formula 1 cars. Despite the widespread use of foils, the skeleton of foil design remains much the same as in the 1930's. The math that defines fluid flow, known as the Navier Stokes equations, has been around since the 18th century but does not have a known solution. Therefore, scores of foils are empirically tested in a wind tunnel or towing tank, and their performance data is stored in catalogs. An engineer searches these catalogs for foil shapes that best satisfy their performance requirements. From there, the whole lifting surface is tested in a wind tunnel to validate performance, and if need be, a new foil shape will be selected. With the rise of computing power, it is possible to approximate solutions to the Navier Stokes equations using programs called CFD's. However, this method still relies on "guessing and checking" to model and validate solutions. There is no guarantee that the foil shape is actually optimal for the given conditions the engineer specifies.

A stochastic algorithm approach called evolutionary computation (EC), could change this design strategy. EC, in this context, would evolve foil shapes to perform as well as possible in a given condition. This project explores the effectiveness of EC in the context of foil design. A *Hill Climbing* EC algorithm is used with a lower fidelity model of foil performance known as *Lifting Line Theory.* Six different scoring mechanisms are used to weigh the many aspects of foil performance. These scoring mechanisms allow an EC algorithm to know what makes one foil "better" than another.

After all six scoring mechanisms are run, each five times, the results are recorded. It appears as if the results of these scoring mechanism creates a *pareto front* for the problem space; a sort of upper bound on foil performance. Moreover, it seems there is a wide range of foil shapes that have identical foil performance.

1 Introduction

A foil is a wing-like surface that generates lift and drag in a moving fluid such as water or air. Foil designs come in all sorts of interesting shapes and sizes, with fascinating contours, winglets, rakes, and sweeps. Their shapes change drastically depending on application. A foil that will work well in a propeller of a cargo ship will have no relation to a wing on a Cessna aircraft, a Formula 1 car, or a supersonic jet. Engineers might design these shapes in several ways. They may mimic biological evolution in a process known as biomimicry. For example, they may study how hawks bend and curl up the tips of their wings, or how waterfowls have raked wings. Or, engineers may look to other human designed foils and try to make slight improvements on them. In either case, they may add some feature to the wing shape that can improve efficiency, but there is no guarantee that the foil shape is actually optimal.

Throughout much of the 20th century, foil shapes were developed by creating specifications for a foil: Rules about how it should perform in certain conditions. Then a foil with the desired characteristics would be selected from a catalog of foils whose performance has been empirically derived, for example, from a wind tunnel. Once a foil has been selected, it would be fitted onto a scale model, and tested in a wind tunnel or a towing tank. From there, design iterations would involve redesigning, and building the model until it has adequate performance. Now, in 2018, computers have improved much of this process, but the skeleton of this design process remains the same. Instead of building models and empirically testing them in wind tunnels, Computational Fluid Dynamics solvers (CFDs) solve partial differential equations governing fluid flow to simulate a wind tunnel or towing tank. Unfortunately, these partial differential equations, known as the Navier Stokes equations, have no known solution, so CFDs use a tremendous amount of computational power to approximate solutions to them. All in all, CFDs reduce design iterations considerably, dropping iteration times from days or weeks, to minutes. However, CFD's cannot tell a designer what is the best foil shape for a condition, and so the "guess and check" method of design still applies.

Evolutionary Computation can have the power to change this. Evolutionary Computation (EC) is used when the a mathematical model behind a problem is complex or even impossible to solve. In this context, a designer would use an EC algorithm by giving it target performance that a foil must meet in certain conditions, and the algorithm will "evolve" a foil with those performance benchmarks, or at least as close as it can get. In EC, a population of randomly created individuals is tested by the model. The individuals (representations of foils in this case) with the best performance score (fitness) are propagated to the next generation. This project employs a simple EC algorithm known as a hill climbing algorithm. In a hill climbing algorithm there are only two individuals in a population: a parent, and a randomly altered, or mutated, child of the parent. If the child performs better against the model than the parent, it goes on to be the parent in the next generation. If the parent performs better than the child, the parent remains the parent in the next generation. By using EC, engineers can get tailor made foil shapes that are made to perform well at an exact condition given by the engineer.

This paper explores the effectiveness of EC in the context of foil designs. Section 2 of this paper will cover background concerning how this project represents foil shapes, the model this project uses instead of the Navier Stokes equations, and what the hill climbing algorithm is. Section 3 will cover the experimental setup, including how the algorithm mutates and scores individuals. Sections 4 and 5 will cover the data received in this project, and a discussion of the results. Sections 6 and 7 will cover conclusions and future work.

2 Background

2.1 Foil Representation — Background

In order to represent foil shapes in a compact way, we must understand important geometric properties of foil shapes that are used by common foil representations. These properties serve as good geometric reference points, but are also directly related to foil performance. The following diagram shows these geometric properties.

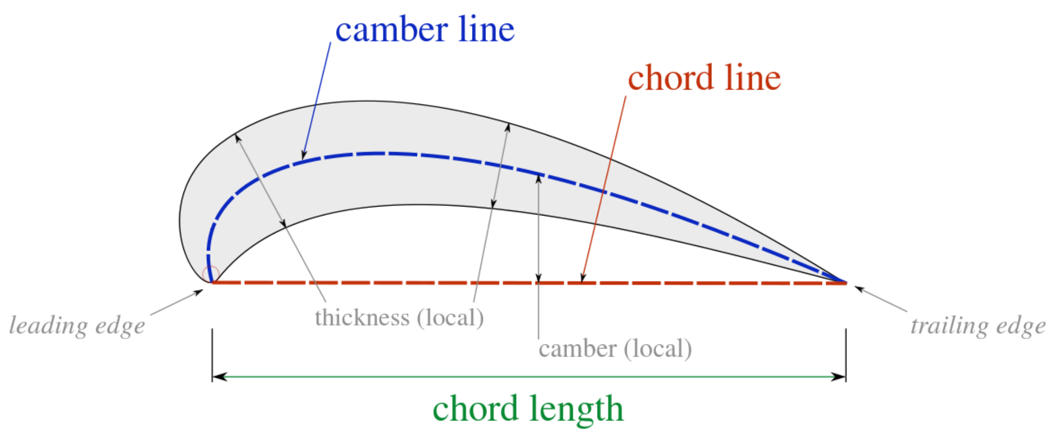


Figure 1: Common geometric properties of foil shapes. Taken from [1]

The red dashed line in the Figure 1 is known as the *chord line* or *chord*. This line connects the leading edge of the foil to the trailing edge. All quantitative measurements of the foil are made in reference to the length of the chord, normally as a percentage. The dotted blue line is known as the *camber line.* This line runs through the foil such that any point on this line is equidistant from the upper and lower surfaces of the foil when measured from the perpendicular of the camber line at the point. The term *camber*, is used describe how far the camber line deviates from the chord line. When a foil has zero camber, i.e., the camber line and chord line are the same, a foil is said to be *symmetrical*. Finally, a *thickness distribution* specifies how the thickness of the foil gradually increases and decreases over the length of the foil to give it its signature teardrop shape that is key for aerodynamic performance.

2.2 Foil Representation — NACA Series

NACA series foils were created by the National Advisory Committee for Aeronautics (NACA), the US government agency that predated NASA. NACA series foils provide a means of standardization and classification of foil shapes such that they can be shared and represented easily. Despite the fact that the NACA series foils were created in the 1930s, they are still commonly used to describe foil shapes. This project employs the NACA 4 digit series foils, the simplest of the NACA series in which a foil shape can be represented by three numbers across four digits. The three numbers used in the NACA 4 digit series encompass the most important geometric features of the foil in regard to aerodynamic performance: The maximum camber of the foil, maximum camber position, and the maximum thickness of the foil.

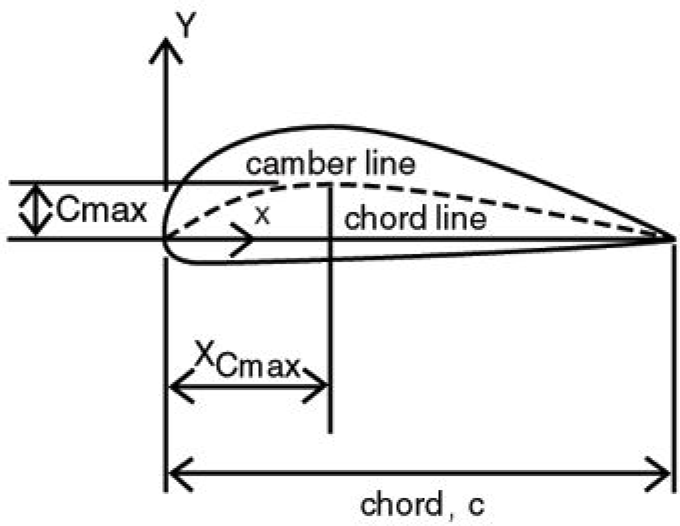


Figure 2: Geometric Properties of NACA 4 Digit Foil. Taken from [2]

The maximum camber, referred to as C_{max} in the figure above, is defined as the shortest distance from the maxima of the camber line to the chord line. The maximum camber position, X_{cmax} in the figure is the distance from the leading edge, along the chord line, to where the perpendicular from the chord line meets the maximum camber.

A NACA foil is formatted as follows: **NACA MPXX**, where *M* is the maximum camber in hundredths, *P* is the maximum camber position in tenths, and *XX* describe the maximum thickness in hundredths with two digits. For example, a NACA 2109 foil with a 1 meter chord will have a maximum camber of 2% the length of the chord (2cm), 10% of the way down the chord line (10cm), with a maximum thickness of 09% of the chord length (9cm).

2.3 Model Specifications

In the context of this project, the model must be able to take environmental conditions and a foil representation as inputs, and return useful metrics about the foil's performance. Aerospace engineers use many metrics to gauge a foil's performance; this project will use lift and drag. It is important to note that theoretically calculating lift and drag is a very difficult mathematical task. The model used in this project is simple when compared to commercial grade modelling software, however it still has many components. Algorithm 1 outlines the major steps in our estimation of life and drag.

The different components of Algorithm 1 will be explained in detail below, but the basic steps are:

- *Thin Aerofoil Theory (TAT)* is used to estimate the *two-dimensional coefficient of lift* of a foil design ($C_{l\text{-}2d}$)
- *Lifting Line Theory (LLT)* is used to estimate the *three-dimensional coefficient of lift* ($C_{l\text{-}3d}$) and *coefficient of drag* ($C_{d\text{-}3dI}$)
- These coefficients are then used to estimate the actual lift and drag.

Algorithm 1 Model (env, foil)

$C_{l-2d} \leftarrow$ TAT (foil, env.α)
C_{l-3d}, $C_{d-3dI} \leftarrow$ LLT (foil, env.α, C_{l-2d})
Lift $\leftarrow$ LiftEq(env, C_{l-3d})
Drag $\leftarrow$ DragEq(env, C_{d-3dI})
return *Lift*, *Drag*

2.4 Lift and Drag

As a foil moves through a liquid, such as air or water, the liquid will exert two forces on the foil. The first force, *drag*, acts in the same direction as flow of the liquid, pushing the foil back in the direction of the fluid. The second force, *lift*, acts *normal,* or perpendicular to the direction of the liquid, this lifts the foil up. If a slight, positive *angle-of-attack* is introduced, the lift and drag will both rise. An angle-of-attack (AOT) is the angle between the chord line of the foil, and the flow of the fluid.

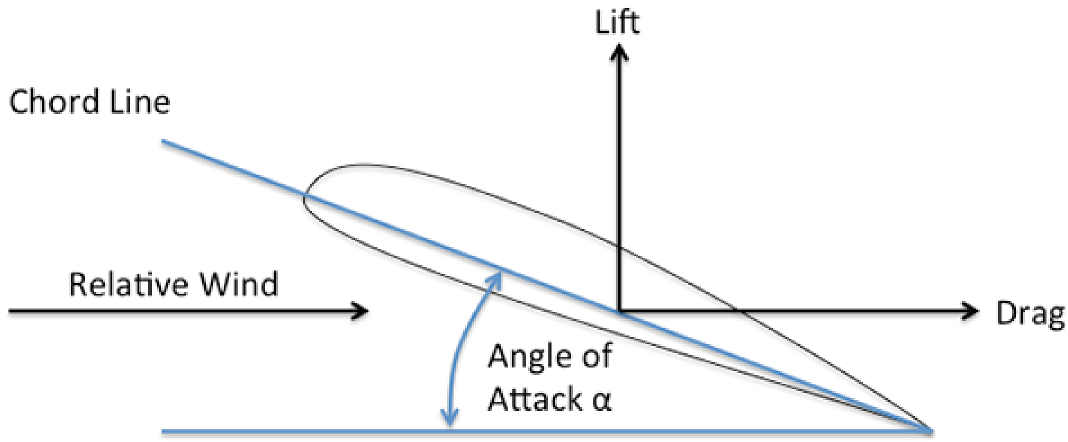


Figure 3: Forces on a foil. Taken from [3]

The the inputs for this model are a foil shape, represented as a NACA 4 digit foil, and environmental conditions such as the fluid velocity (V_∞), the fluid density (ρ), and angle-of-attack (α). These environmental factors are used in the following equations for lift and drag in two dimensions:

$$L = 1/2\, \rho\, V_\infty^{\ 2}\, C_{l-2d}\, A \qquad \text{Equation 1}$$

$$D = 1/2\, \rho\, V_\infty^{\ 2}\, C_{d-2d}\, A \qquad \text{Equation 2}$$

In these equations, L and D represent the lift and drag of a foil. A is the *cross sectional area* of the foil shape. The cross sectional area of the foil is the area between the upper and lower surfaces of the foil. $C_{l\text{-}2d}$, and, $C_{d\text{-}2d}$, are referred to as the *coefficient of lift,* and the *coefficient of drag* respectively. These coefficients describe how the geometric properties of a foil produce lift and drag when exposed to a particular angle-of-attack.

For reasons that will be discussed in the subsequent sections, it is also important to understand lift and drag equations for three-dimensions. The equations are very similar:

$$L = 1/2\, \rho\, V_\infty^{\ 2}\, C_{l-3d}\, S \qquad \text{Equation 3}$$

$$D = 1/2\, \rho\, V_\infty^{\ 2}\, C_{d-3d}\, S \qquad \text{Equation 4}$$

The difference between these lift and drag equations and the two dimensional lift and drag equations is that the coefficients of lift and drag are now three dimensional coefficients. In addition, the cross sectional area of the foil is replaced with S, the *Wing Area.* Wing Area describes the area that a three dimensional wing takes when looked down on from above. For example, a typical fighter jet, when looked down on from above has a trapezoidal wing shape; its wing area would be the area of that trapezoid.

2.5 Models Used

The predominant equations that govern fluid flow are called the *Navier Stokes* equations. These equations define characteristics of fluids in any conditions, and effectively model a wind tunnel or towing tank. Unfortunately, the Navier Stokes equations are a set of partial differential equations with no known solution. In order to use them, an extraordinary amount of computational power must be used to approximate their solutions. The algorithms that are commonly used approximate these equations are known as Computational Fluid Dynamics Algorithms (CFD's). CFDs are very accurate, but are expensive and time-consuming to compute. This presents a challenge to the project because Evolutionary Computation needs to be able to run the model hundreds of thousands of times, very quickly, and efficiently. Therefore, a CFD can not be used to do this sort of modelling.

Instead of CFDs, this project uses a combination of two models that were developed in the early 20th century to quickly model lift and drag for foils, albeit, with less accuracy than a CFD. The first model implemented, known as *Thin Airfoil Theory* (also referred to as *TAT* in Algorithm 1) takes in environmental factors and a foil shape. It returns the two dimensional coefficient of lift of the foil shape (C_{L-2d}). Then, the two dimensional coefficient of lift, along with the environmental conditions, and the foil shape are given to another model known as *Prandtl's Lifting Line Theory* (*LLT* in Algorithm 1). Lifting Line Theory will return three dimensional coefficients of lift and drag (C_{L-3d}*, and* C_{D-3di}) for an elliptically shaped foil having the NACA foil as its cross section. Both of these theories are described in more detail below.

2.6 Thin Airfoil Theory

Thin Airfoil Theory[4] was developed during the 1920s by an engineer named Max Munk[5]. The idea of the model is that lift comes from circulation occurring over the foil as the foil is exposed to a moving fluid. Thin Airfoil Theory assumes that the fluid is inviscid (frictionless), incompressible (subsonic), and irrotational. Moreover, Thin Airfoil Theory has strict assumptions about the foils in question. It assumes that the foil is a two-dimensional cross section, but more notably, that the foil is infinitely thin. The combination of the fluid assumptions and the zero thickness condition allows lift to be modelled from a *vortex sheet* along the camber line. This model has been used in foil design for a significant time, and can provide reasonably accurate results under normal conditions. However, because of the inviscid fluid assumption, the fluid has no friction, and thus no drag or drag coefficient can be modelled by Thin Airfoil Theory. This is a problem considering that designs that optimize for high lift are extremely different from designs that optimize for low drag. This is where Lifting Line Theory allows for the extrapolation of a coefficient of drag.

2.7 Lifting Line Theory

Lifting Line Theory[6] was developed by Prandtl during the 1920s[7] and is an extension of Thin Airfoil Theory to three dimensions. The reason why Thin Airfoil Theory does not work on its own in three dimensions is because of what is known as induced drag at the wing tips. Foils work by creating high pressure underneath the surface, and low pressure above the surface. However, at the wing tips, high pressure air from underneath the lifting surface can bleed into the low pressure air above. This bleed creates what is known as an induced vortex. Without any shaping of the wingtips, induced drag can form a substantial part of drag that aircraft face. By using Lifting Line Theory, the model can estimate drag.

In addition to modelling wingtip vortices, Lifting Line Theory models how lift is distributed over three dimensional shapes. Lifting Line Theory dictates that the best lift distribution for improving lift and reducing drag is an elliptical lift distribution. The three-dimensional foil shape that produces an elliptical lift distribution is an ellipse. The farther a wing deviates from an elliptical wing shape, the higher the induced drag. In order to estimate the coefficient of induced drag, but reduce computational expense, the three-dimensional shape of all evolved foils is an ellipse. Lifting Line Theory will take in the two dimensional coefficient of lift from the Thin Airfoil Theory, the two dimensional cross section shape (represented by a NACA 4 digit foil), and environmental conditions, and will return three-dimensional coefficients of lift and drag.

2.8 Deriving Lift and Drag

Lifting Line Theory returns three-dimensional coefficient of lift and drag values. However, the goal of the model is to return lift and drag. Therefore, the three-dimensional coefficients of lift and drag are used in equations 3 and 4 with the environmental factors to produce values for lift and drag. The model then returns these lift and drag values.

2.9 Evolutionary Computation and Hill Climber Algorithm

Evolutionary Computation (EC) algorithms are stochastic algorithms inspired by biological evolution. The premise is that a population of potential solutions is tested against a model of the problem, the best individuals move onto the next generation and produce “offspring”. Over many generations the performance against this model will improve until either a target performance is obtained, or the algorithm reaches a fixed number of generations. This project employs a *hill climbing algorithm,* an EC algorithm that mimics the process by which simple single cell organisms evolve asexually through genetic mutation . The following figure shows pseudocode for the hill climbing algorithm.

Algorithm 2 Hill-Climber

```
S ← some initial candidate solution
repeat
    R ← Tweak(Copy(S))
    if Quality( R ) > Quality( S ) then
        S ← R
until S is the ideal solution or we have run out of time
return S
```

Algorithm 1: Hill-Climber[8].

At generation 0 of the run, a random individual is created, this individual is called the *parent of generation 0.* A new individual is created by making a random change to parent; this individual is called the *child of generation 0*. From there, the two individuals are scored against the model. The individual with the best score becomes the parent of generation 1, and the process continues until the algorithm reaches a pre-set number of generations, or until the individual's performance exceeds some target value. A hill climbing algorithm was chosen for this project because of its simplicity. There is little overhead in implementing it, and the results are easy to interpret. All in all, a hillclimber serves as a good proof of concept.

3 Experimental Setup

3.1 Hill Climber Implementation

In this project, the pseudo code for the hill climber algorithm remains the same, however there are important details to note about how it is configured for runs. The purpose of the runs in this project is to explore the effectiveness of an EC algorithm in the context of foil design. For this reason, the hill climber evolves for a fixed number of generations rather than stopping when a foil exceeds certain performance benchmarks. All experiments run for 3,000,000 generations. This value is should be big enough to capture most if not all of the evolutionary progress possible by a hill climber in this context.

3.2 Mutation Function

The mutation function takes in an individual, represented as a NACA 4 digit foil, and modifies its *M* and *P* values. The mutation function does not change the thickness value of the foil because of the assumption that Thin Airfoil Theory makes about the foil being infinitely thin . The *M* value is restricted to values between 0 and 9.5 (9.5% of the chord length). *P* is restricted to values between 0 and 9 (90% of the chord length). The mutation function will vary the parents' *M* and *P* values within a range that is defined by *M* (or *P*) plus or minus an exploration factor. The exploration factors for both *M* and *P* are set to 2. If this range exceeds the restrictions on either *M* or *P,* the offending section of the range will be capped at the restricted value.

Algorithm 4 Mutation (parent)

```
                                        ▷ parent is a NACA 4 Series Foil
m-exp-factor ← 2
m-upper-bound ← upper-bound-check(m-exp-factor + parent.M, 9.5)
m-lower-bound ← lower-bound-check(m-exp-factor + parent.M, 0)
child ← newIndv()
child.M ← random(m-lower-bound, m-upper-bound)
return child
```

For example, suppose Algorithm 4 is called with a parent NACA91XX. The range of possible values for the child before *upper-bound-check* and *lower-bound-check* are, for M, from 7 to 11, and for P, from -1 to 3. The upper and lower bound check functions will cap the ranges such that the new individual's M and P values will stay within the predefined restrictions of M being between 0 and 9.5, and P being between 0 and 9. After these upper and lower bound checks, the resulting range will be for M, between 7 and 9.5, and for P, between 0 and 3. A new child will be created with M and P values within this range.

3.3 Problem Setting

In order for an individual to have a drag component with this model, it must exist in three dimensions. As mentioned in the Lifting Line Theory section, all individuals will have an elliptically defined, three dimensional shape in order to save computational expense. The lift and drag values given back by the model will be derived from three dimensional coefficients of lift and drag created from Lifting Line Theory.

In an effort to evolve solutions for practical problems, historical examples are used. When studying Lifting Line Theory, one aircraft emerges as the embodiment of Lifting Line Theory in application: the Supermarine Spitfire. The spitfire is a world war two era fighter designed in the epoch of Lifting Line Theory's use in aircraft design. In particular, the spitfire featured elliptically defined wings with a thin foil-cross-section (The root of its wing was a NACA 2213 foil). These features gave it extraordinary speed, and maneuverability for its time. It is important to note that the wing was for structural reasons, not perfectly elliptical, and had many geometric details that are not captured in this model. Moreover, the introduction of a fuselage creates many aerodynamic issues that are too complicated to account for in this model. However, general geometric properties of the spitfires wings, and performance characteristics can be used to create problem settings that are rooted in real aircraft performance.

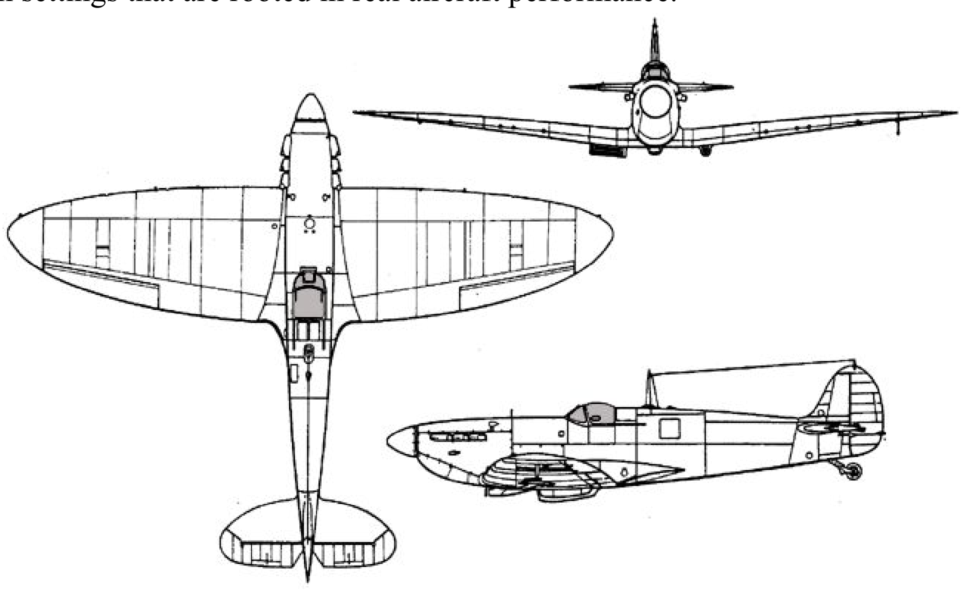


Figure 4: Supermarine Spitfire Diagram. Taken from [9]

In particular, individuals will share the same *span* and *root chord* length. Span is defined as the distance from one wing tip to the other, while the root chord length is the length of the chord at the centerline of the aircraft. The span for the spitfire was 11.24 meters[11], with a root chord of 2.67 meters[10]. All individuals will have an elliptical shape derived from these two properties. Moreover, the wing was inclined 2.1 degrees with respect to the fuselage[10]. Even as the aircraft flies perfectly horizontally, the wing experiences an angle of attack of 2.1 degrees [10].

All tests in this paper focus on running a "spitfire like" elliptical wing in a relatively low airspeed, and low altitude configuration. Foils in these tests are exposed to a fluid moving at 30 meters per second (67 miles per hour) at 2.1 degrees of angle of attack. The fluid has the same density as it would near sea level, 0.7708 $\frac{kg}{m3}$. Lift and Drag will be measured in Newtons.

3.4 Scoring Functions

Lift and Drag often rise and fall in concert with one another. This presents challenges in design because a foil that has high lift, will often have high drag (and visa versa). It is up to the scoring function to weigh how important high lift is opposed to low drag. Because a hill climber needs a single numerical answer for a score, lift and drag must be somehow combined into one number that represents how "good" a foil is in a given condition. The following scoring functions accomplish this in different ways. It is important to note that in this project, a higher score represents better performance.

3.4.1 Sufficient Lift

This scoring function acts most like an engineer might. In sufficient lift scoring, the individual's lift is compared to a *minimum lift*. If the individual's lift is lower than this minimum lift, the individual receives a score of zero. However, if the individual's lift is higher than the minimum lift, the individual's score will be set to the largest 32 bit integer minus the individual's drag. This scoring function incentivises foils that have just the amount of lift an aircraft will need to be stable, but from there, minimize drag.

3.4.2 Sufficient Drag

The sufficient drag model compares an individual's drag to a *maximum drag*. The maximum drag must be small enough such that the spitfire has enough airspeed to not stall when the engine is at maximum thrust. Like sufficient lift scoring, if the individual's drag is greater than the maximum lift, the individual receives a score of zero. If the individual's drag is less than the maximum drag, the score will be set to the individual's lift value. Sufficient drag scoring incentivises foils to have as much lift as possible, while still staying below the maximum drag.

3.4.3 Linear Combination Scoring

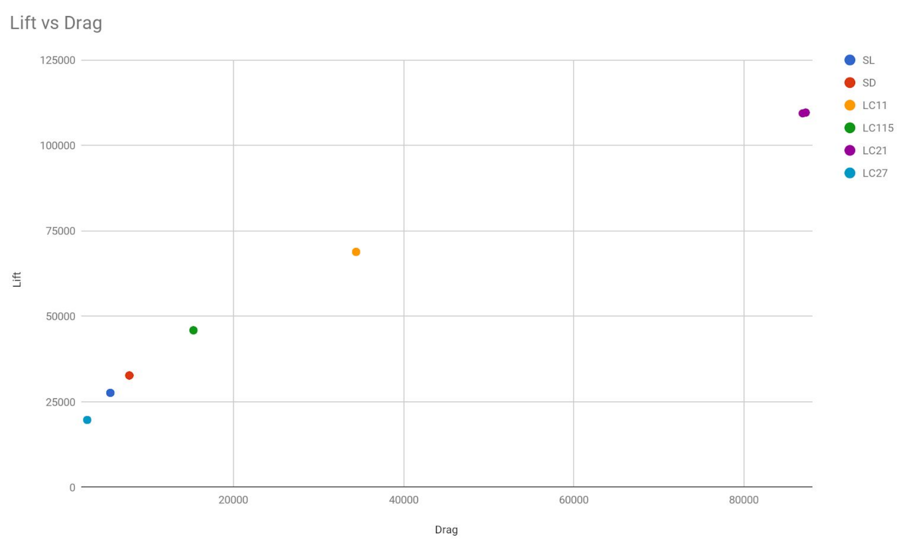
Linear Combination Scoring is normally the simplest way of doing scoring when more than one objective (lift, drag, etc…) needs to be combined into one score. In this case, the lift and drag values both have their own weights and score is simply:
$Score = LiftWeight * Individual.Lift - DragWeight * Individual.Drag$

The lift and drag weights stay the same throughout an entire test. This project tried several different lift and drag weights to see to what degree the algorithm optimizes for high lift or low drag. This test employed linear combination scoring with lift and drag weights respectively: (1, 1), (1, 1.5), (2, 1), and (7, 2).

4 Results

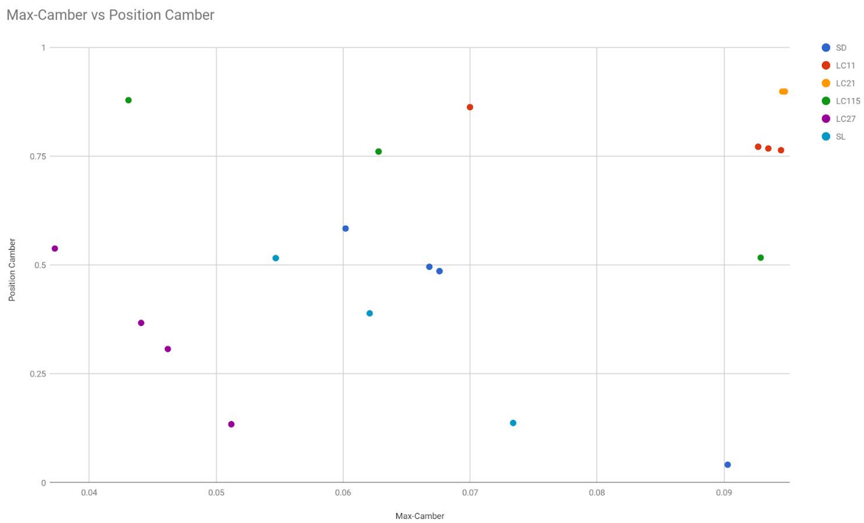
Each scoring function was tested against the slow speed, low altitude configuration five times. The following graphs show the results from these runs. Overall there are 6 different scoring functions, resulting in 30 tests in the dataset. Each run outputs information about lift, drag, *M* and *P* values, 2d and 3d coefficients of lift and drag, and generation numbers for each parent in a run. The following graphs show the aggregated data for this particular set of runs.

4.1 Lift vs Drag



This graph displays the lift and drag of each of the five runs of each scoring function. Even though it may appear as though there are only six data points, all five results for each scoring function are there, but so tightly grouped that they cannot be differentiated.

4.2 *M* vs *P* Values



This graph shows the m and p values for each of five runs in each of the scoring mechanisms. The *M* and *P* values are scaled between 0 and 1 for easier interpretation.

5 Discussion

5.1 Lift vs Drag — Pareto Front

When looking at the Lift vs Drag graph, it is apparent that all five runs of each scoring function have such similar lift and drag values that it is impossible to distinguish them in the graph. These groups of points seem to indicate that all five runs converge on at least a local optima for that scoring function, and possibly the global optima. More interestingly, all of the best solutions from these points seem to create a "logarithmic like" trend line.

Since the objective of this algorithm is to create solutions with high lift and low drag, individuals want to score closer to the top left of the Lift vs Drag graph. It is easy to create solutions that place below the logarithmic trend line, but it is hard to find solutions that place above it. These runs may indicate that this logarithmic line is the *pareto front* of the problem space. The parent front is the boundary of how well a solution can do in this model. Knowing the pareto front of the problem space helps designers know, for a given condition, exactly how ambitious their design objectives can be.

5.2 *M* vs *P* — Linear Trend

It would be reasonable to expect that five runs, each with the same scoring functions, all with the same lift and drag values would have similar *M* and *P* values. However, this is far from the case. The *M* and *P* graph seems to indicate that the top five solutions to these problems have a linear correlations for each group. It would be interesting to see if an individual made up of *M* and *P* values on one of these lines, would have the same lift and drag values as the other individuals on the lines.

Based on these trend lines, a variety of solutions can be created that have the same lift and drag. This is useful because different shapes might be better for performance in other conditions, or might have better structural or maintenance characteristics. Even if the perfect foil shape, in terms of structure or performance in other conditions, doesn't exist for a certain lift or drag. Choosing a solution on the trend line made by running this algorithm could provide a foil with better shape characteristics that still performs on the pareto front.

6 Conclusion

Currently all results are speculation, much work still needs to be done to confirm these findings. More runs will have to be done within each of the scoring functions to confirm the existence of linear trends between these groups. More scoring functions will have to be generated to confirm the "logarithmic like" trend line, and to prove it is a pareto front. The points on this possible pareto front must be modelled to create a regression. Furthermore, it will have to be tested that individuals on the same linear trendline have the same performance with lift and drag.

All in all, it is important to note that these results do not use navier stokes equations, and do have error in their results. Thin Airfoil Theory and Lifting Line Theory do not account for many complex characteristics of fluid flow, and approximations made by them must be taken with a grain of salt. However, it would be interesting to see how the trends found in these two graphs do hold up in models that use Navier Stokes equations.

7 Future Work

There are many areas of growth for this project, and many opportunities to improve algorithmic efficiency, accessibility, and to widen applications. First of all it is prudent to validate the results created in the current state of the project. The steps to do validate these results are mentioned in the conclusion. However, gains can be made in both discovery and efficiency by considering the following evolutionary algorithms to replace the hill climbing algorithm.

- **Lexicase selection:** This algorithm uses a population of individuals per generation. The best individuals of one generation are taken and tweaked by many genetic operators. Instead of just mutating individuals, two parent's genetic information are combined to create one or two children in the next generation (akin to sexual reproduction). The results would be an entire generation of winning solutions with a broad degree of diversity between solutions.
- **MAP Elites:** Just last summer (summer 2017), the Map Elites algorithm has been used to create a variety of diverse solutions for foil design[12]. This work was published in GECCO (The Genetic and Evolutionary Computation Conference)in the summer of 2017. The map elites algorithm is a population based EC algorithm that enforces diversity among its generations by splitting up the solution space into regions. In each region, a population based EC algorithm discovers several local optimas and hopefully a global optima for the region. Implementing this

algorithm would serve as a useful an exercise in reproducing work done by researchers in the field.
- **NSGA II:** The NSGA II algorithm takes a different approach to finding useful solutions. Instead of solving solutions to find local and global optima for a problem space, NSGA II works entirely by advancing a pareto front. As it runs, it identifies a pareto front comprised of the best current solutions. It then creates individuals that it believes may fill sparser gaps in the pareto front.

Beyond just algorithmic improvements, the project could improve from better modelling. [12] improved modelling in their EC algorithm by using a gaussian process to approximate lift and drag for foil shapes. It is also possible to create modells by using the vast series of empirically tested foil performance data to create a model using Neural Networks.

Furthermore, the emphasis of the project could change from evolving a foil for the best performance in one set of conditions, to finding the best foil shape for a variety of conditions an airfoil might encounter in flight. Overall, there is great area for research like this to grow in prevalence.

References

[1] Wikipedia. 2018. Wikipedia: Chord (aeronautics). In *Wikipedia, The Free Encyclopedia*. Retrieved March 23, 2018.
[2] AKiTi. A Description of the NACA 4-Digit System. Retrieved from http://www.akiti.ca/NACA4Bkgrnd.html
[3] Code7700. 2017. Angle of Attack. Retrieved from https://www.code7700.com.
[4] MIT. 2008. Fluids — Lecture 2 Notes. Retrieved from https://web.mit.edu.
[5] Wikipedia. 2018. Wikipedia: Thin Airfoil Theory. In *Wikipedia, The Free Encyclopedia*. Retrieved March 23, 2018.
[6] University of Notre Dame. 2015. Lifting Line Theory and Finite Wings Lab. Retrieved from https://www3.nd.edu.
[7] Wikipedia. 2018. Wikipedia: Lifting Line Theory. In *Wikipedia, The Free Encyclopedia*. Retrieved March 23, 2018.
[8] Sean Luke. 2013. *Essentials of Metaheuristics,* Lulu, second edition, available at http://cs.gmu.ed/~sean/book/mataheuristics.
[9] Wikipedia. 2016. Wikipedia: Supermarine Spitfire. In *Wikipedia, The Free Encyclopedia*. Retrieved March 23, 2018.
[10] J. A. D. Ackroyd. 2016. *The Aerodynamics of the Spitfire,* Journal of Aeronautical History, available at
https://www.aerosociety.com/media/4953/the-aerodynamics-of-the-spitfire.pdf .
[11] John F. Guilmartin. 2014. *Spitfire*, Encyclopaedia Britannica, available at https://www.britannica.com/technology/Spitfire.
[12] Adam Gaier, Alexander Asteroth, Jean-Baptiste Mouret. Data-Efficient Exploration, Optimization, and Modeling of Diverse Designs through Surrogate-Assisted Illumination. *Genetic and Evolutionary Computation Conference (GECCO 2017)*, 2017, Berlin, Germany. 2017.