AIAA SciTech Forum
6-10 January 2020, Orlando, FL
AIAA Scitech 2020 Forum

10.2514/6.2020-1296

Airfoil Optimization Using Cross-Entropy Method on Conformal Mapping

Bradley Rafferty[*]
Stanford University, Stanford, CA 94305, USA

An algorithm has been developed to optimize airfoil performance through iterative design via conformal mapping. The current version of the algorithm produces notable results for maximizing lift-to-drag and for minimizing drag. Test cases for a Reynolds number of 200,000 and an increased lift-to-drag from 55.0 to 71.0 (angle of attack of 2 degrees) and reduced coefficient of drag from 107.4 counts to 60.7 counts (angle of attack of 0 degrees), where each objective function optimization was run separately. The algorithm is rapid in that it can design and evaluate an airfoil within approximately 0.2 to 0.3 seconds. The optimization routine is proved to show promise in being extended to a wider range of objective functions such as, for example, maximizing laminar run or determining an optimal flap deflection angle. Limitations of the current implementation center on the simplicity of the elected conformal mapping method and the lack of rigorous constraints. Future improvements to the optimization routine are poised to be integrated readily.

Nomenclature

C_D	=	coefficient of drag	y	=	coordinate in airfoil (transformed) plane
C_L	=	coefficient of lift	z	=	airfoil (transformed) plane
c_χ, c_η	=	center of circle in ζ-plane	α	=	angle of attack
I_q	=	identity matrix of dimensions q x q	ζ	=	circle (original) plane
L/D	=	lift-to-drag ratio	η	=	coordinate in circle (original) plane
m_{elite}	=	number of elite samples	θ	=	normal distribution parameters
n	=	number of airfoils designed per batch	μ	=	mean of normal distribution
N	=	number of batches per run	σ	=	variance of normal distribution
r	=	radius of circle in ζ-plane	χ	=	coordinate in circle (original) plane
x	=	coordinate in airfoil (transformed) space			

Superscripts

$\rightarrow$	=	vector

Subscripts

0	=	initial value

I. Introduction

Airfoil design is critical to the performance of many aerodynamic objects such as airplanes, Formula-1 racecars, and wind turbines. The design determines the aerodynamic behavior of the resultant wing, be it the drag, lift, pitching moment, etc. To design an airfoil for an application of interest typically requires expert-level knowledge and intimate understanding of fluid mechanics and aerodynamics. In that regard, the field of airfoil design has a large knowledge barrier-to-entry. Because of this barrier, much of the research on airfoil design—in the literature or otherwise, especially algorithmic airfoil design optimization—relies on iterating on the shapes of airfoils that already exist, i.e. starting from a "seed" or baseline airfoil (e.g. Ref. [1]). These seed airfoils are often sourced from databases such as those listed in Refs. [2, 3] or from other well-documented and well-known production or research airfoil designs. While this seed-based approach is important and productive in its own right, it leaves much to be desired in terms of exploring the design space for potential global or even local optima for a given application.

[*] Graduate Student, Department of Aeronautics and Astronautics, AIAA Member

Additionally, it is most common that geometry generation within airfoil optimization routines relies on spline-based methods or parameterization methods such as B-Splines and PARSEC, respectively [4]. As such, one may perceive a general lack of novel approaches to airfoil design. The subject implementation of algorithmic design utilizes conformal mapping instead. In general terms, conformal mapping is a transformation from one geometric shape (often one that is relatively convenient and simple to define) to a more complicated geometric shape. While rigorous conformal mapping techniques can be quite exacting to develop and implement, the subject implementation offers a relatively simplified approach with minimal user input and expertise required.

II. Approach

This section will outline the architecture of the optimization routine and the formal expression of the problem formulation.

A. Optimization Routine

The subject approach to optimizing an airfoil design for a given setpoint flow condition and objective function is quite simple. To enable efficient design for this particular investigation, conformal mapping is employed in order to reduce the dimensionality of the design space compared to, for example, spline-based design. One particular conformal mapping scheme of interest is the Joukowsky transform [5]. This transform maps a circle to an airfoil shape; the outputs of the transform are the x-y coordinates of the resultant airfoil design in Euclidian space. This method is elegant in that this particular formulation only requires only three inputs: the coordinates of the center of a circle, and the radius of that circle, r. Hence, only three parameters needs to be specified to output an airfoil design. Modifying spline parameters, discrete airfoil coordinates directly in Euclidean space, or even certain parameterizations (e.g., PARSEC requires 12 inputs) could prove to be a much higher-dimensional and more complex approach. Eqs. (1-5) below outline the Joukowsky transform.

$$\vec{z} = \vec{\zeta} + \frac{1}{\vec{\zeta}} \tag{1}$$

$$\vec{\zeta} = \vec{\chi} + i\vec{\eta} \tag{2}$$

$$\vec{\chi} = rsin(\vec{\beta}) + c_\eta \tag{3}$$

$$\vec{\eta} = rcos(\vec{\beta}) + c_\chi \tag{4}$$

$$\vec{z} = \vec{x} + i\vec{y} \tag{5}$$

In the above equations, the airfoil is generated in the z-plane after being transformed from the circle defined in the ζ-plane. The center points of a given circle are (c_χ, c_η) and $\vec{\beta}$ is an angle vector in the range $[0, 2\pi]$ whose spacing determines the number of points on the circle and hence the airfoil. The resultant coordinates of the airfoil are retrieved from Eq. (5): x and y are the real and imaginary components of z. The inputs required to solve Eqs. (1-5) are c_χ, c_η, and r. The mapping can be simplified even further to require only two inputs, c_χ and c_η, by imposing certain constrains as discussed in Section II.B of this paper.

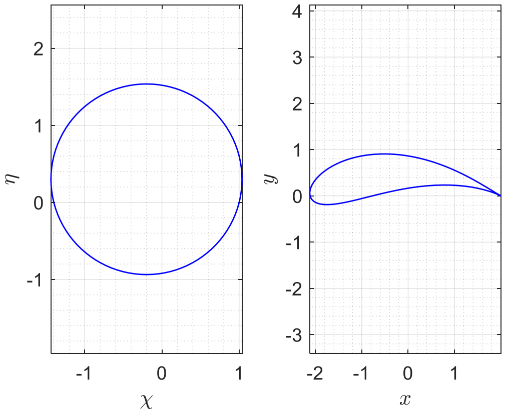

Figure 1 *Example Joukowsky transform*

The three Joukowsky transform inputs are initialized to some value, and then are updated with each iteration of the optimization algorithm. The updated values are determined by a proposal distribution generated within the cross-entropy method—for this work, a normal distribution was chosen. The cross-entropy method is an algorithm that refines the design space by updating the chosen proposal distribution based on the highest-performing samples of the current distribution, also known as the elite samples. The cross-entropy method is chosen for its elegance, simplicity of implementation, and its stochastic nature. An advantage of a stochastic method for this design problem is that it may better explore the design space and avoid local minima—i.e., it could perhaps produce non-intuitive airfoil geometries that perform similarly to or better than well-documented designs. See Figure 2 for an example of elite parameters evaluated from a distribution and the resulting update to the multivariate normal distribution.

Once a design-point airfoil is generated, it is analyzed using XFOIL—an open-source, panel-method analysis tool that solves for the pressure distributions of subsonic, isolated airfoils. XFOIL is chosen because it is a powerful, computationally-inexpensive, well-documented [6], and widely-used tool with many functionalities. The only required inputs to XFOIL are the x-y airfoil coordinates, Re, and α. Though, other flow conditions such as Mach number can be set if desired. See Figure 3 for an example output pressure distribution plot from XFOIL.

A shell is developed to automate the entire design and evaluation process (built in MATLAB, but easily replicated in Python, etc.). Through this shell, XFOIL is automatically fed input commands through the command prompt. Its outputs—the objective function evaluations—are retrieved and the resultant array of objective function evaluations is used to rank the performance of the airfoils for that batch. The elite designs are determined by the parameter m_{elite}, which dictates the quantity of highest-performing designs to be used for the next distribution. The mean and variance of the conformal mapping parameters for these elite airfoils are used to update the proposal distribution within the cross-entropy method. This new, updated distribution is then sampled and evaluated sample-by-sample, as before, and the cycle continues until convergence criteria or a prescribed limit is met. Currently, a limit is set on the number of airfoil designs per batch (100) and the total number of batches (8). The only inputs required are $\vec{\theta}$, n, N, m_{elite}, Re, α, and any constraints. Figure 4 illustrates the flow of the entire optimization routine.

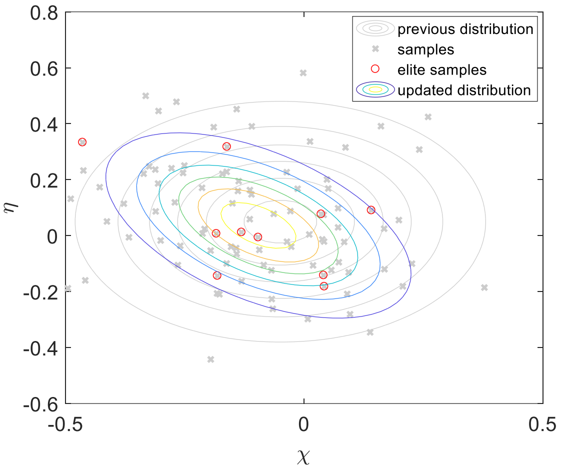

Figure 2 *Example proposal distribution update*

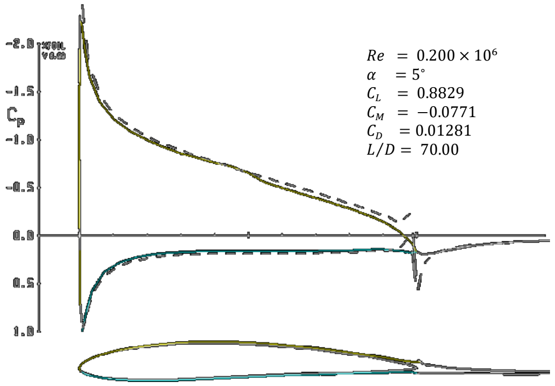

Figure 3 *Example XFOIL pressure plot*

B. Problem Formulation

The optimization routine employed and the tools utilized offer flexibility in the choice of objective function. For this particular study, the author focuses on two separate, single-objective-function cases: maximizing *L/D* (minimizing -L/D) and minimizing C_D. The formal problem statement for both is identical:

$$\underset{c_\chi, c_\eta, r}{\text{minimize}} \quad f\left(c_\chi, c_\eta, r, Re, \alpha\right). \tag{6}$$

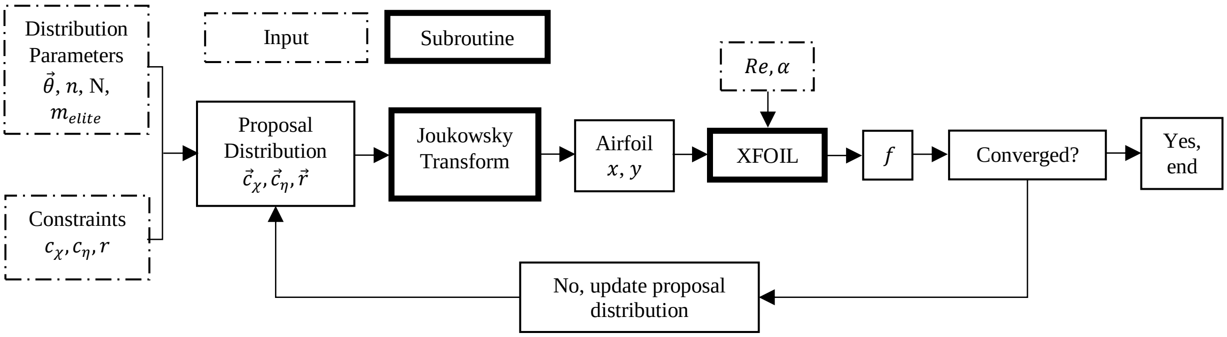

Figure 4 *Architecture of the optimization routine*

The current implementation of the algorithm allows the solution to investigate the results of exploring the design space without imposing constraints. This constraint-free approach leverages the simplicity of allowing the evaluation tool, XFOIL, to determine infeasible designs by failing convergence. Failed designs are assigned poor objective function evaluations in order for the algorithm to learn which regions of the design space to avoid. While this approach requires less forethought from the designer, there is a significant added cost in computation time when XFOIL fails to converge. Ideally, infeasible designs could be screened out before passing through for evaluation in XFOIL. Though, even seemingly-sensible designs can fail convergence in XFOIL if the combination of airfoil design and flow conditions cause large flow separation. A relatively simple tweak to the problem formulation in Eq. (6), as seen in Eq. (7-9), can promote the generation of well-formed airfoils.

$$\underset{c_\chi, c_\eta}{\text{minimize}} \quad f\left(c_\chi, c_\eta, Re, \alpha\right) \tag{7}$$

$$\text{subject to} \quad r = \left\| 1 - c_\chi, c_{eta} \right\|_2 \tag{8}$$

$$r > \left\| -1 - c_\chi, c_{eta} \right\|_2 \tag{9}$$

Note that the $\|x_1, x_2\|_2$ notation denotes the L2 norm of x_1 and x_2 (also known as the Euclidian norm). The formulation in Eq. (7-9) can force the circle to intersect the point $\zeta = 1$ (Eq. (8)), and then check whether the circle encloses the point $\zeta = -1$ (Eq. (9)). If one of these constraints are not satisfied, then the airfoil will be malformed (contain intersections, more than one cusp, etc.) and hence it can be discarded before evaluation in XFOIL. These constraints also mean that the objective function only requires two input parameters to generate an airfoil as opposed to the three in the formulation expressed in Eq. (6). The results in this paper are of the formulation expressed in Eq. (6), to better understand how the optimization routine may or may not learn the constrained design space instead.

C. Parameter Selection

The inputs to the algorithm and to XFOIL are listed in Table 1. Both N and n were chosen to permit rapid convergence of the algorithm without sacrificing exploration from too few samples in the multivariate normal distribution. To permit sufficient sample distribution from one proposal distribution to the next, m_{elite} was chosen to be 10 percent of the total samples per batch. Choosing too few m_{elite} could result in poor exploration and choosing too many could result in poor convergence characteristics of the cross-entropy method. The initial conformal mapping parameters were selected based on trial-and-error in order to begin the algorithm in a productive design space in terms of the generation of sensible airfoil geometries. Sufficient exploration of the design space is enabled by both an appropriate m_{elite} and a large enough σ. The flow parameters Re and α are chosen as singular values representative of low-Reynolds-number conditions, however the algorithm can easily be adapted to evaluate multiple Re and α for each airfoil design.

Table 1 *Various parameters input to the algorithm*

N	n	m_{elite}	$[c_{\chi_0}, c_{\eta_0}, r_0]$	σ	Re	α
8	100	10	[-0.05, 0.05, 1.1]	0.05*I_3	200000	0

III. Results

This section will overview the results of the two cases of interest that are outlined in Section II.B.

A. Maximizing *L/D*

For the case of (Re, α) = (200000, 2°), the L/D increased from 56.0 to 66.0 from batch 1 to batch 8, an improvement of 17.9 percent. To reiterate, there are no seed airfoils in this method—all airfoils are generated within the optimization routine. As expected for an airfoil with high-performing L/D, the algorithm converges to a thin and highly-cambered airfoil. In terms of optimizing the objective function, the author considers the algorithm quite successful. However, it is clear that not all resultant airfoil designs may be practical for all applications. The progression of optimal airfoil design for each batch is illustrated in Figure 5.

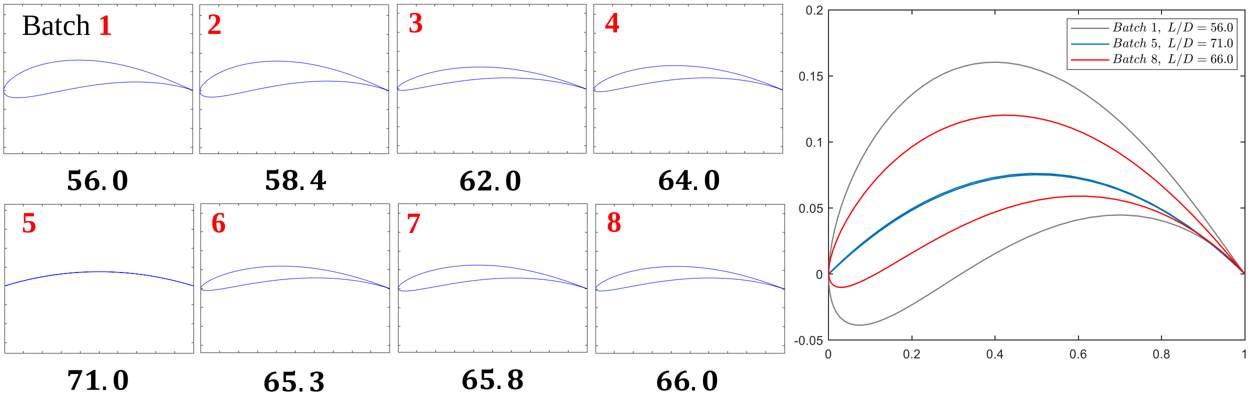

Figure 5 *The top elite designs from each batch of* $\boldsymbol{L/D}$ *maximization and three designs overlaid on the same plot*

It is interesting to note that the algorithm discovers what could be near a global optimum within the design space at the fifth batch. Due to the stochastic nature of the cross-entropy method, the next elite designs within batches 6 through 8 do not perform as well as batch 5, but indeed are improvements compared to batch 1. The total time to design and evaluate all 800 airfoils is 320.7 minutes. This averages to 24.1 seconds per airfoil, which accounts for significant delays due to non-converged XFOIL evaluations that raise the average time considerably as well as secondary operations such as moving and discarding files. The number of non-converged evaluations tends to decrease with each new batch, i.e. the algorithm learns to avoid designs that are infeasible because non-converged designs are penalized with an objective function evaluation of $L/D = 0$. This trend is illustrated in Figure 6. Evaluation failures within XFOIL are most often due to poorly-defined trailing edges, e.g. edges which are too sharp or perhaps even intersecting. Separate functions within the tool can detect and remove certain cases of intersections, but, in general, the tool does not have the ability to completely screen out malformed or otherwise undesirable shapes that may be produced by the conformal mapping stage and may fail evaluation in XFOIL. To that end, imposing more rigorous geometric constraints on the airfoil would improve the non-converged evaluation rate. It is found that properly-evaluated airfoils require approximately 0.2 to 0.3 seconds to be designed and evaluated.

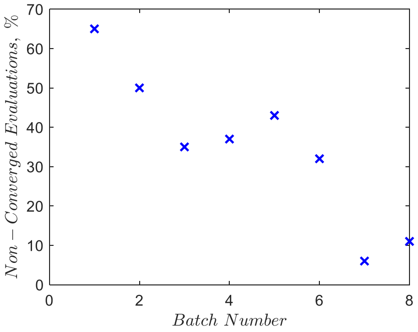

Figure 6 *Failed evaluations per batch*

Compared to well-known high-lift, low-Reynolds-number airfoils, the algorithm-designed airfoils perform quite well. Figure 7 illustrates a comparison of XFOIL pressure distribution plots. The algorithm-designed airfoil from batch 8 exceeds the *L/D* of E420 and nearly matches that of the S1223 reference airfoils at $L/D = 66.0$, 55.1, and 66.1, respectively. Interestingly, the lift of the algorithmically-designed airfoil is lower than that of both the reference airfoils, however its drag is considerably lower than both as well. The pressure distribution plots indicate that the laminar run for the algorithm airfoil is much longer than both reference airfoils, which explains the significantly-improved drag performance over the references. It may be of interest to the reader that E420 and S1223 are both

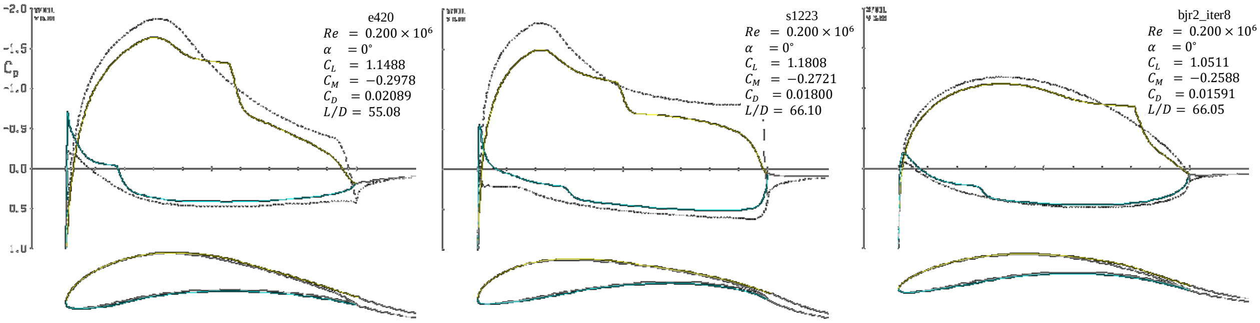

Figure 7 *Pressure distribution plots of two reference airfoils and the algorithm-designed airfoil for f = L/D*

popular designs for wing packages on hobbyist and collegiate Formula-1 racecar competitions. The former is a design by Richard Eppler and the latter by Michael Selig.

B. Minimizing C_D

Drag minimization is also performed at the flow conditions (Re, α) = (200000, 0°). Again, the algorithm produces notable results and reduces C_D from 107.4 counts to 60.7 counts between batches 1 and 8, an improvement of 43.5 percent (note: 1 count is an increment of 0.0001 in C_D). As expected, the algorithm converges to a thin and quasi-symmetric design. The observations of the results in this case are similar to those of maximizing *L/D*: the algorithm optimizes the objective function quite well. Furthermore, the algorithm rapidly converges to a much-improved objective function, although not all resultant airfoil designs may be useful in practice. The top-performing airfoils of each batch are plotted in Figure 8.

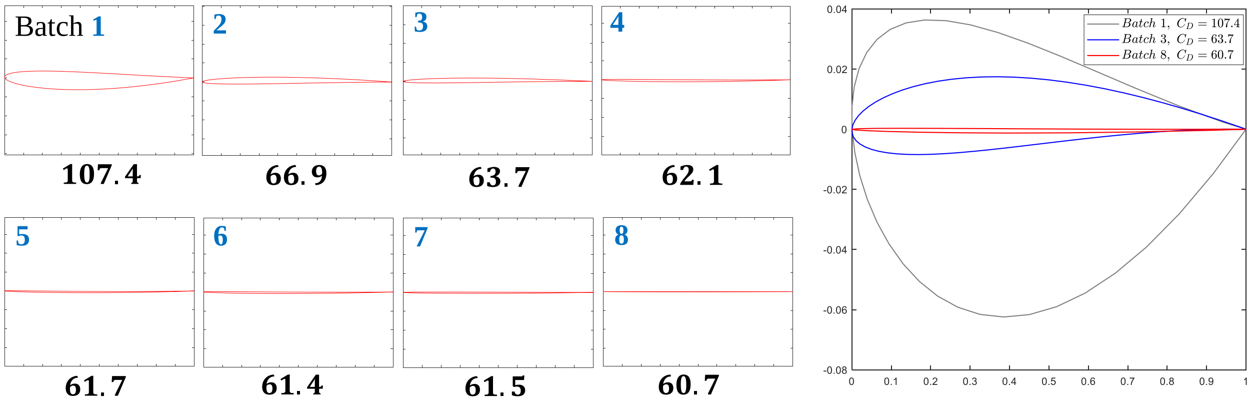

Figure 8 *Elite designs for drag minimization from each batch and three designs overlaid on the same plot*

Due to the stochastic nature of the algorithm and the characteristics of this particular conformal mapping method (the Joukowsky transform), the airfoils are not guaranteed to be perfectly symmetric—although, a global optimum would be for this case of drag minimization at $\alpha = 0$. There are at least two options to force a symmetric design for future implementations of this method: (1) simply mirror one side of any given airfoil design about its mean camber line, or (2) use a symmetric airfoil transform such as the Tsien Transform [7].

The total time to design and evaluate all 800 airfoils for these batches is 55.2 minutes or 4.14 seconds per airfoil. Unlike the previous test case of Section III.A, the number of non-converged evaluations seems to remain relatively constant across the eight batches as seen in Figure 9. Interestingly, the total run time is faster than that of the L/D minimization case, indicating that each non-converged case does not incur a fixed penalty to runtime, but rather each case is unique in terms of how quickly XFOIL handles an infeasible design.

The results from this case are compared to the well-documented low-drag airfoils S9033 and naca0006 in Figure 10. The reader may be interested to know that the S9033 is used on the vertical stabilator of a remote control airplane made by Black Hawk. Of course, naca0006 is of the acclaimed four-digit series by the National Advisory Committee for Aeronautics. The algorithm-designed airfoil performs very well at nearly 100 counts less drag compared to both reference airfoils, although it is considerably thinner.

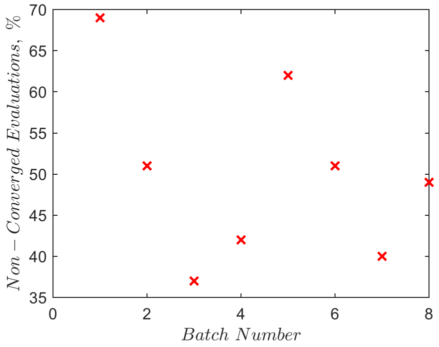

Figure 9 *Failed evaluations per batch*

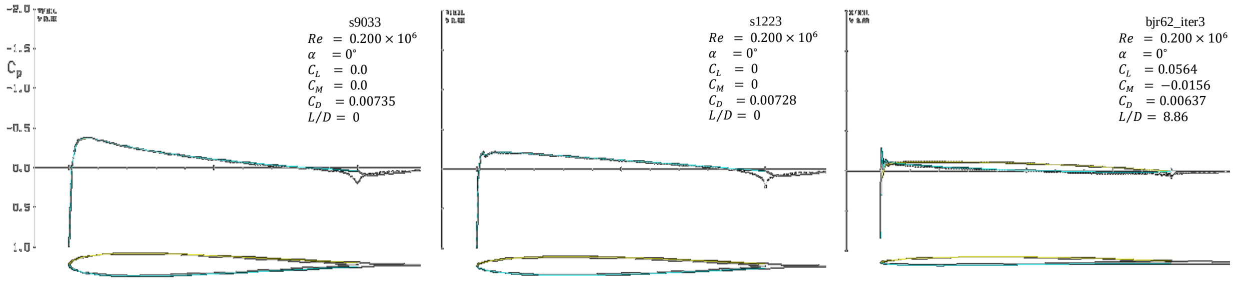

Figure 10 *Pressure distribution plots of two reference airfoils and the algorithm-designed airfoil for $f = C_D$*

IV. Conclusions and Future Work

The algorithm proves to successfully optimize two separate airfoil objective functions through iterative, stochastic design via conformal mapping. The algorithm designs, evaluates, and converges relatively quickly in the two test cases that are discussed. Furthermore, the optimization routine can readily be adapted to optimize other objective functions or even consider a multiobjective optimization problem through, for example, a weighted method. Possibilities to improve and expand this work include: (1) evaluating multiple *Re* and α cases for each design, (2) incorporating more sophisticated conformal maps (e.g., Ref. [8]), (3) applying the same methodology to spline-based designs or a parameterization method such as PARSEC, (4) adding more rigorous constraints to the problem formulation (e.g., on camber, thickness, etc.), and (5) comparing the cross-entropy method to another stochastic optimization problem such as simulated annealing. Above all, the work encourages a promising foundation and test-bed for a noteworthy airfoil optimization approach.

References

[1] Krantz, P., and Hedman, S. G., "Airfoil Optimization," *Journal of Aircraft*, Vol. 23, No. 5, May 1986, pp. 355–356 doi: 10.2514/3.56773

[2] Selig, M., "UIUC Airfoil Coordinates Database," *UIUC Airfoil Coordinates Database* [online database], URL: https://m-selig.ae.illinois.edu/ads/coord_database.html [retrieved 1 May 2019]

[3] Eppler, R., Airfoil Design and Data, Springer-Verlag, New York, 1990

[4] Sripawadkul, V., Padulo, M., and Guenov, M., "A comparison of Airfoil Shape Parameterization Techniques for Early Design Optimization," AIAA Paper 2010-9050, September 2010.
doi 10.2514/6.2010-9050

[5] Joukowsky, N.E., "Über die Konturen der Tragflächen der Drachenflieger", *Zeitschrift für Flugtechnik und Motorluftschiffahrt*, Vol. 1, pp. 281-284, 1910, and Vol. 3, pp. 81-86, 1912

[6] Drela, M., "XFOIL," *Subsonic Airfoil Development System* [website], URL: https://web.mit.edu/drela/Public/web/xfoil/ [retrieved 1 May 2019]

[7] Selig, M., and Maughmer, M., "Multipoint Inverse Airfoil Design Method Based on Conformal Mapping," *AIAA Journal*, Vol. 30, No. 5, May 1992, pp. 1162–1170

[8] Tsien, Hsue-shen, "Symmetrical Joukowsky airfoils in shear flow," *Quarterly of Applied Mathematics*, Vol. 1, No. 2, July 1943, pp. 130-148