Airfoil Optimization Based on Neural Network and Derivative-free Algorithms

Haolin Liu
Mechanical Engineering
Carnegie Mellon University
haolinl@andrew.cmu.edu

Songlin Liu
Mechanical Engineering
Carnegie Mellon University
songlin1@andrew.cmu.edu

Wenzhao Ye
Mechanical Engineering
Carnegie Mellon University
wenzhaoy@andrew.cmu.edu

Chenxi Wang
Mechanical Engineering
Carnegie Mellon University
chenxiwa@andrew.cmu.edu

Jingfeng Liu
Mechanical Engineering
Carnegie Mellon University
jingfenl@andrew.cmu.edu

Ge Lv*
Robotics Institute
Carnegie Mellon University
glv@andrew.cmu.edu

Abstract

Airfoil contour shape optimization is a fundamental part in the field of aerodynamic design. In this project, the optimized airfoil contours are obtained by conducting genetic algorithm, using a pre-trained convolutional neural network (CNN) as the fitness function evaluating airfoil's lift-to-drag ratio (C_L/C_d). Based on the raw coordinates data from UIUC airfoil dataset, cubic interpolation and a method of Bézier curve fitting are implemented under certain geometric constraints to generate smooth airfoil contour curve. Then a method of gradient-free local search is used to speed up the optimizing process and a certain number of airfoil individuals are chosen to reproduce new contours for the coming generations [3]. After trying optimizing airfoils with genetic algorithm, the method of simulated annealing is also used to improve individual's C_L/C_d. Results show that the performance of airfoil is averagely improved by 3.8%; specially in the case of genetic algorithm, we found that contributed by pre-trained CNN and local search, the general process of optimization is accelerated by approximately 33%.

KEYWORDS: Airfoil optimization, CNN, lift-to-drag ratio, derivative-free algorithms.

1. Introduction

Airfoil optimization is an important part of aerodynamic design, increasingly drawing lots of attention in the field of aeronautical engineering. Many researchers have worked on several methods that can optimize this special 2D geometric shape to reach the aerodynamic performance that they desire. Among all the methods, the derivative-free algorithms are the most popular ones that are known as their high computation efficiency. The way to conduct these iterative methods can be easily achieved by high performance computer and nice structured programs [3].

In order to fix attentions to certain points, two concrete algorithms - genetic algorithm and simulated annealing algorithm - are chosen to be implemented in this project. With pre-trained convolutional neural network (CNN) as evaluation function, each airfoil individual's lift-to-drag ratio (C_L/C_d) is calculated and taken as representative of corresponding aerodynamic property (shown as Figure 1). In this project, different algorithms are proved to work better in different situations, and with proper parameters and initial definitions, improvement can be observed quite apparently.

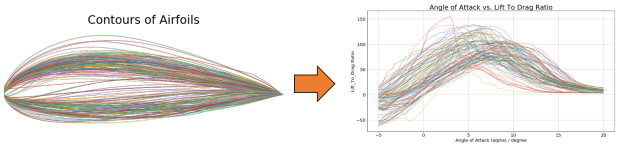


Figure 1: Airfoil's contour - evaluation function

To design applicable airfoil contours, several geometric constraints are necessary to be applied so that optimization can be directed to a reasonable domain. In this project, apart from normalization and contour's first and second order continuity (except two points at the leading and trailing ends), all airfoils are smoothed to different corresponding Bézier curves, and the leading edge of the airfoil are further enforced to be convex. These constraints can help optimization process yield better results and save optimization time from unnecessarily generating bad airfoil contour shapes. What's more, using CNN as evaluation function, as well as combining algorithms with non-gradient local search method, is also proved to be a good way of shorten-

ing iteration time. More algorithm implementation details will be revealed and explained in this project report.

This project is derived from another project finished by *Haolin et al* concerning training CNN for predicting airfoil's C_L/C_d. Both these two projects are on the basis of *Xfoil*, an aerodynamic numerical analysis software used to generate data groundtruth. To further remove confusion related to the situation this project is truly studying, the Reynolds number is set to a constant of $400,000$ and the Mach number is set to 0.0, referring to low speed, incompressible airflow. And to make the aerodynamic performance be only the function of airfoil's shape, the angle of attack is fixed at 0. The situation fits for the air-drone with low running speed. Without special notation, all the lift-to-drag ratios, as well as all airfoil individuals, are calculated and optimized under this special case.

2. Related Work

Among the approaches that try to solve airfoil optimization problems, Gardner and Selig [3] find optimal airfoil shapes by utilizing a genetic algorithm through manipulation of the velocity distribution. The airfoil geometries used in their approach are generated by an inverse method from velocity distribution parameters, and *Xfoil* is used to determine fitness values especially for symmetric airfoil individuals. Hacioglu and Ozkol [1] introduce vibrational genetic algorithm and its implementation to inverse airfoil design. The vibration mutation and vibration crossover are used in their approach. The research shows a good results in passing over local optimums and making the convergence faster. Zhao and Wang *et al* [2] optimize the suction control for airfoil design using multi-island genetic algorithm (MIGA). The optimal suction location on the upper airfoil surface is found through minimize both the airfoil drag and suction requirement. Thier results show a better performance of MIGA over traditional GA in maintaining population diversity and an overall higher calculation speed.

Similar with implementations of genetic algorithm in plane airfoil design, researches show promising performances in wind turbine airfoil optimization problems. Huque and Zemmouri *et al* [6] perform multi-objective design optimization using Elitist Non-dominated Sorting Genetic Algorithm (NSGA-II). Pareto optimal set of six different 2D airfoil profiles are generated by NSGA and each Pareto optimal solution represents a different compromise between design objectives. Ram *et al* [5] develop a multi-objective genetic algorithm to optimize the tip region of turbine airfoil blades in regard to roughness insensitivity. Similar to *Gardner and Selig*'s approach, Orman and Durmu [4] perform a study to find the best airfoil representation scheme which gives the best lift to drag ratio in a large design space for the ideal aerodynamic design optimization. From next section, the modified genetic algorithm used in our optimization project is introduced.

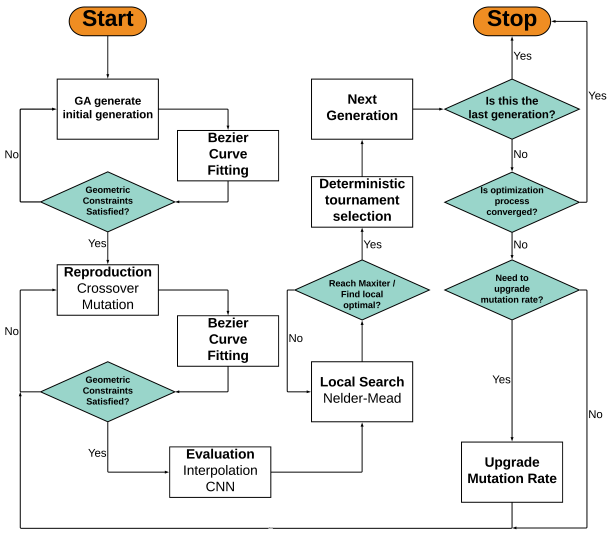


Figure 2: Flowchart of genetic algorithm

3. Genetic Algorithm

3.1. Introduction

Genetic algorithm (GA) is one of the most commonly studied evolutionary algorithms (EA) in the field of optimization and operation research. Inspired by Darwin's theory of evolution, genetic algorithm can generates high-quality optimization solutions by iteratively conducting selection, crossover and mutation processes. With different goals, genetic algorithms can be implemented in different ways and structures. In this project, the control points of airfoil's contour curve (fitted Bézier curve) are chosen to be variables, and the optimization process will be conducted under certain geometric constraints. Structure of genetic algorithm is shown as Figure 2.

3.2. Implementation

The overall process of implementation is: First, GA will generate initial population (first batch parents) by Bézier curve fitting, which will help smooth the contour curve; secondly, GA needs to make sure that all contours are subjected to geometric constraints; third, certain crossover and mutation, which are also called reproduction, will be conducted among population; after reproduction, the fitness value of each airfoil individual shall be calculated, and local search algorithm will also be implemented afterwards; with evaluated fitness values, certain number of airfoils will be chosen for next generation. This process will repeat until the algorithm reaches its stopping criterion. Each part will be

discussed as follows.

3.2.1 Initial population

Based on the UIUC online airfoil dataset, the raw coordinate data of airfoils is used as data of control points for Bézier curve fitting. Since the airfoils from online dataset have already been optimized, they cannot be directly used as initial population; instead, new airfoil contours generated from raw dataset can be regarded as new airfoils and can be utilized as initial parents. The raw data is firstly cubic interpolated by 800 points, with leading edge denser and trailing edge sparser (shown as Figure 3); then 200 points would be picked out among them as control points for Bézier curve fitting.

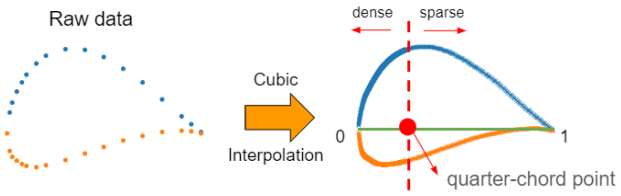


Figure 3: Cubic interpolation

In order to make genetic algorithm work efficiently, the size of population is restricted to 100 airfoil individuals, and all the curve fitted from raw dataset is controlled to $100th$ order Bézier curve.

3.2.2 Geometric constraints

As an airfoil, curves on both upside and downside of contour should be smooth and continuous, which refers to curve's first and second order continuity. Bézier curve fitting will be implemented here to achieve smooth and continuous contour curve (shown as Figure 4). This constraint can be removed at the points on leading and trailing ends of the contour. In order to make the result derived from CNN is more accurate, bias between dataset need to be removed. Since the CNN is trained with dataset that seldom contains airfoils with concave leading edge, the samples put into the neural network shall also be without concave leading edge. Similarly, since CNN only get inputs from image with resolution of 128×128, the airfoil should be normalized to a certain reasonable range. Therefore, the airfoil individuals should be subjected to "convex leading edge" constraint and should lie in the range of $[0, 1]$ along x-axis.

3.2.3 Reproduction

There are two main parts of reproduction, crossover and mutation. For the crossover, this project chooses the strategy called "binary swapping", which means only upside or downside of the contour will be swapped between airfoils. This crossover strategy can generally ensure the continuity at the leading and trailing points (and also avoid generating weird airfoil shape). The crossover rate is set to 0.5. For mutation, it will be restricted within a range controlled by a parameter called "mutation ratio", meaning the mutation range will be equal to it multiplied by the maximum range along the axis. The ratio is equally set to 0.001 for both x and y axis, and the overall mutating rate is set to 0.01.

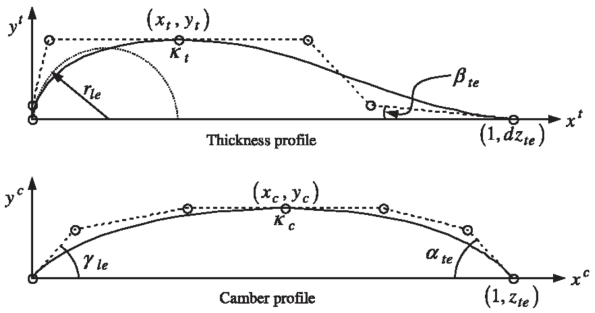


Figure 4: Bézier curve fitting in genetic algorithm

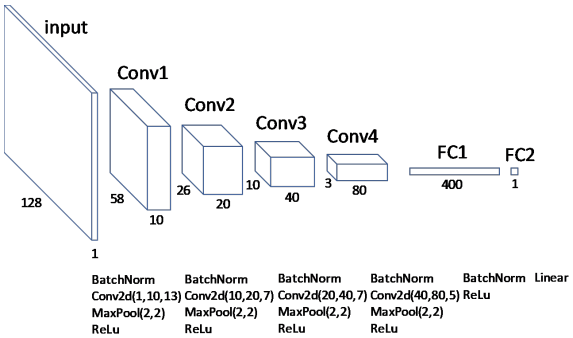


Figure 5: Structure of CNN

3.2.4 Evaluation

This project takes pre-trained CNN as fitness function, which means the calculated C_L/C_d from CNN will be considered fitness value. The structure of CNN is shwon as Figure 5. After each airfoil is evaluated with an fitness value, all of these values will be put into *softmax* function in order for the better selection in next step.

3.2.5 Selection

After obtaining fitness vector, the same number of individuals will be selected with probability related to their

fitness values. In this project, the strategy of deterministic tournament selection is used, and the tournament size is set to 20.

3.2.6 Local Search

In order to reduce the iterating time for optimization, the method of gradient-free local search is implemented. Before conducting reproduction in the next round, certain amount of airfoils will be randomly picked out from current population pool; setting these individuals as starting points, the nelder mead algorithm will then be conducted to find the nearby local maximum on fitness function. After the local maximum is found or the maximum iteration number is reached, the result individual and corresponding fitness value will replace the individual at starting point, and then be returned to the original population pool. The process of implementing local search algorithm is shown as Figure 6.

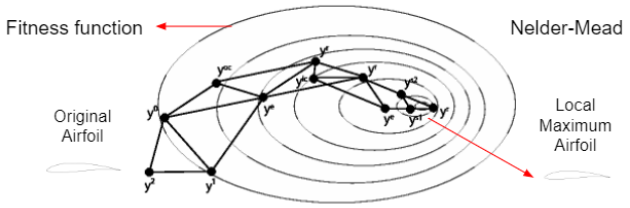


Figure 6: Local search - Nelder-Mead algorithm

3.2.7 Criterion

To end the algorithm, there are several thresholds: if the results show convergence in 5 continuous generations, the algorithm will automatically stop; or in a more usual case, if the number of iterations has reached the maximum number of 150 generations, the algorithm will be forced to abort. During the optimization process, the global best candidate and its corresponding lift-to-drag ratio shall be recorded and maintained until the end of algorithm. No matter how the optimization process ends, the final global best contour will be taken as the final optimization result.

4. Simulated Annealing

4.1. Introduction

Simulated annealing (SA) is a probabilistic technique for approximating the global optimum of a given function. Specifically, it is a metaheuristic to approximate global optimization in a large search space for an optimization problem. The name and inspiration come from annealing in metallurgy, a technique involving heating and controlled cooling of a material to increase the size of its crystals and reduce their defects. This notion of slow cooling implemented in the simulated annealing algorithm is interpreted as a slow decrease in the probability of accepting worse solutions as the solution space is explored. Accepting worse solutions is a fundamental property of metaheuristics because it allows for a more extensive search for the global optimal solution.

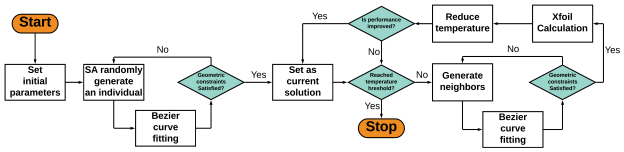


Figure 7: Flowchart of simulated annealing algorithm

In general, the simulated annealing algorithms work as follows. At each time step, the algorithm randomly selects a solution close to the current one, measures its quality, and then decides to move to it or to stay with the current solution based on either one of two probabilities between which it chooses on the basis of the fact that the new solution is better or worse than the current one. During the search, the temperature is progressively decreased from an initial positive value to zero, and make the probability of moving to a worse new solution progressively change towards zero.

4.2. Implementation

4.2.1 Modeling

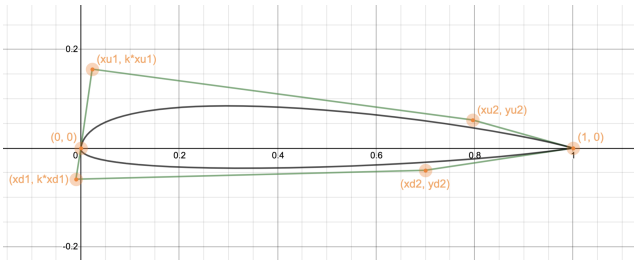


Figure 8: Modeling Airfoil with 2 Cubic Bézier Curves

To simplify the model of airfoil, two cubic Bézier curves are used to represent the shape of the airfoil. Both curves start from $(0,0)$ and end $(1,0)$. This is requirement of XFOIL which is addressed in next section. A reasonable constraint is to make two curves G_1 continuous at $(0,0)$. Thus the two control points (x_{u1}, y_{u1}), (x_{d1}, y_{d1}) lie on the same line with slope k. G_1 continuity only requires the two tangent vector have the same direction, but they do not have to share the same length as shown in Figure 8
The negative null form of the optimization problem is
Maximize $f(\mathbf{p})$, $\mathbf{p} = [k, x_{u1}, x_{d1}, x_{u2}, y_{u2}, x_{d2}, y_{d2}]^T$, subject to $\mathbf{p} \in \mathbb{R}^7$. $f(\mathbf{p})$ refers to a function that takes the

decision variable **p** and return the lift drag ratio.

4.2.2 Neighbor

The neighbors are generated using a randomized method. Since each element in **p** has different meaning, a learning rate vector **l** is introduced to customized the change of elements in **p**.

$$\mathbf{p}_{new} = \mathbf{p} + (rand() - 0.5) \cdot \mathbf{l}$$

After generating the new decision variable $\mathbf{p}_{new}$, we evaluate the lift drag ratio of the old and new airfoil and get y and y_{new}. In simulated annealing, the fact that accepting bad solution by chance helps lead to the global optima.

$$Prob = rand() - exp(\frac{y_{new} - y}{T})$$

If the value of the equation above is greater than 0, a bad solution is accepted in this iteration. After each iteration, the temperature T is decreased by 10%. So the chance of accepting bad solution is getting smaller.

4.2.3 Fitness

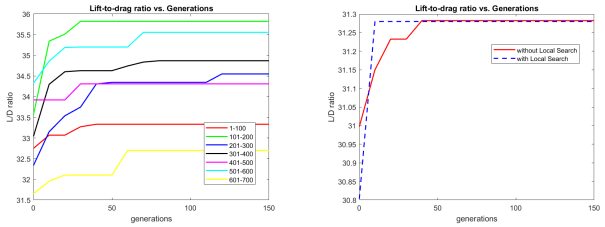
XFOIL is used to calculate the lift drag ratio given the decision variable **p**. XFOIL is an interactive program for the design and analysis of subsonic isolated airfoils. Given the coordinates specifying the shape of a 2D airfoil, Reynolds and Mach numbers, XFOIL can calculate the pressure distribution on the airfoil and hence lift and drag characteristics. XFOIL requires that the coordinates of the airfoil to be in the range of $[0, 1]$. The leading edge of the airfoil lies at $(0, 0)$ and the trailing edge lies at $(1, 0)$.

XFOIL combines a panel method and an integral boundary layer formulation for the analysis of the potential flow around airfoil. Overall, it gives more accurate and reliable results than neural network. However, XFOIL solver might have non-convergence issue, on the contrary, the neural network always gives a result for any given airfoils. Additionally, a maximum number of iterations has to be specified when using XFOIL. This limitation is also a factor that can hinder the speed and accuracy of the solution.

5. Results and Discussion

5.1. Genetic algorithm

Figure 9a shows the results of genetic algorithm optimization, each curve of which represents a optimization process from initial best individual to the global best individual. The curves indicate that airfoil's C_L/C_d can be averagely improved by 3.84%. Also, due to local search algorithm, most of the optimization processes reached their convergence with 100 generations.

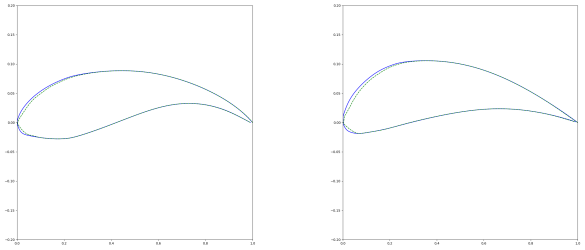


(a) C_L/C_d vs. Generations - Genetic algorithm

(b) Curves with and without local search

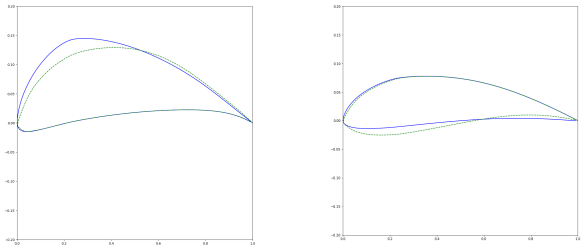
Figure 9: Result curves of Genetic Algorithm

The difference between the initial best airfoil and global best airfoil should not be very large, since the optimization should not change the original airfoil design too much. Figure 10 shows the comparison of initial and optimized airfoil contour curves.



(a) Improved 6.22%

(b) Improved 5.15%



(c) Improved 6.74%

(d) Improved 1.78%

Figure 10: Initial & Optimized airfoil contours

To study the actual contribution of local search, an experiment of conducting optimization process with and without Nelder-Mead algorithm on the same population is tested. The size of testing population is limited to 50 airfoil individuals. The result in Figure 9b shows that the overall iterating time for optimization process is reduced by 33%.

5.2. Simulated annealing

Figure 11 shows the change of lift drag ratio as the number of iterations grows. The vertical axis is the global max-

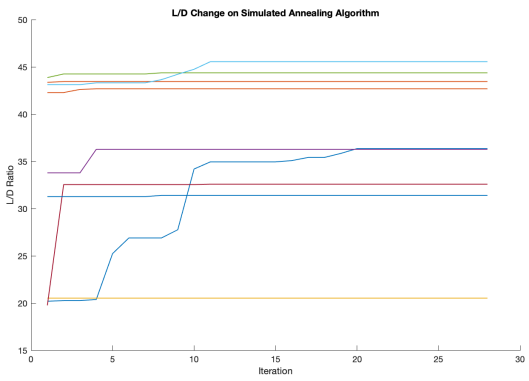
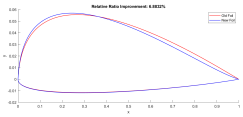
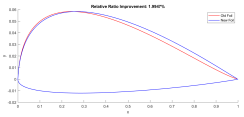


Figure 11: C_L/C_d on Simulated Annealing Algorithm

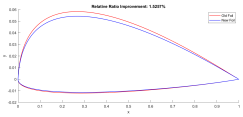
imum lift drag ratio that appeared so far, and the horizontal axis is the number of iterations. Due to the feature of simulated annealing, it is not guaranteed that each iteration will give better lift drag ratio. Many parts of the curve remains the same lift drag ratio because no new maximum appears in these iterations.



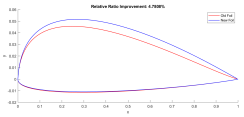
(a) Improved 6.88%



(b) Improved 1.99%



(c) Improved 1.52%



(d) Improved 4.78%

Figure 12: Airfoil L/D Improvement

Figure 12 shows the comparison between the old airfoil and the refined airfoil.

6. Conclusion

This project shows the feasibility of using neural network and derivative-free algorithms to optimize the airfoil's contour and improve their aerodynamic performance. Genetic algorithm and simulated annealing algorithm are both concretely implemented to optimize this special 2D geometry and successfully achieve an average of 3.8% improvement in terms of C_L/C_d. This project uses pre-trained CNN as evaluation function, which slightly improves the optimizing efficiency. Also, this project successfully uses Nelder-Mead algorithm as local search method and combines it with the main genetic algorithm to speed up the optimization process. Result shows that the time spent in optimization iteration has been saved by approximate 33%.

With comparison between genetic algorithm and simulated annealing, the conclusion can be easily got that genetic algorithm is more proper for finding and generating better airfoil individuals among a population pool, while simulated annealing is better for changing and optimizing on a basis of a single individual. Both these two non-gradient algorithms can achieve considerably wonderful results on airfoil optimization.

Acknowledgement

We are specially thankful to the previous co-workers *Zi Li* and *Felix Lu*, who made their contribution on CNN training and dataset optimizing in the last project. We also thank *Gus Brown* and *Louis Edelman*, who provided us the program for converting *Xfoil* tasks to *Matlab* codes. Specially thank to *Ge Lv*, the instructor of the course *Engineering Optimization* in *Carnegie Mellon University*, who taught us basic optimization knowledge and gave us important suggestions for this project, and also, gave us an opportunity to conduct this project on our own.

References

[1] I. O. Abdurrahman Hacioglu. Vibrational genetic algorithm as a new concept in airfoil design. 2002. Aircraft Engineering and Aerospace Technology, Vol. 74 Issue: 3, pp.228-236.

[2] W. C. P. Z. D.J. Zhao, Y.K. Wang*. Optimization of suction control on an airfoil using multi-island genetic algorithm. 2014. Procedia Engineering. 99. 696-702. 10.1016/j.proeng.2014.12.591.ainable Energy, 5 (052007). pp. 1-15. ISSN 1941-7012.

[3] M. Gardner, Ben Selig. Airfoil design using a genetic algorithm and an inverse method. 2003. Department of Aeronautical and Astronautical Engineering University of Illinois at Urbana-Champaign.

[4] E. Orman and G. Durmu. Comparison of shape optimization techniques coupled with genetic algorithm for a wind turbine airfoil. 2016. 2016 IEEE Aerospace Conference, Big Sky, MT, 2016, pp. 1-7.

[5] K. R. Ram, S. P. Lal, and M. R. Ahmed. Low reynolds number airfoil optimization for wind turbine applications using genetic algorithm. 2013. J. Renewable Sustainable Energy 5, 052007 (2013).

[6] D. H. Ziaul Huque, Ghizlane Zemmouri and R. Kommalapati. Optimization of wind turbine airfoil using nondominated sorting genetic algorithm and pareto optimal front. 2012. International Journal of Chemical Engineering, vol. 2012, Article ID 193021, 9 pages, 2012.