Comparison of ACO and GA Techniques to Generate Neural Network Based Bezier-PARSEC Parameterized Airfoil

Waqas Saleem[1], Riaz Ahmad[2], Athar Kharal[3], Ayman Saleem[4]

W. Saleem is with School of Mechanical and Manufacturing Engineering National University of Sciences, Pakistan (Phone: +923224362442, e-mail: waqas_jeral@hotmail.com

R. Ahmad is currently director research at CIE Building, Research Directorate National University of Sciences and Technology, Pakistan (e-mail: dresearch@nust.edu.pk)

A. Kharal is currently Associate Professor at College of Aeronautical Engineering National University of Sciences and Technology, Pakistan (atharkharal@gmail.com)

A. Saleem is with College of Aeronautical Engineering National University of Sciences and Technology (aimen1173@hotmail.com)

Abstract— This research uses Neural Networks to determine two dimensional airfoil geometry using Bezier-PARSEC parameterization. Earlier, Ant Colony Optimization (ACO) techniques have been used to solve combinatorial optimization problems like TSP. This work extends ACO method from TSP problem to design parameters for estimating unknown Bezier-PARSEC parameters that define upper and lower curves of the airfoil. The efficiency and the performance of ACO technique was compared to that of GA. The work established that ACO exhibited improved performance than the GA in terms of optimization time and level of precision achieved. In the next phase, Neural Network is implemented using Cp as input in terms of C_l, C_d and C_m for learning and targeting the corresponding Bezier-PARSEC parameters. Neural Networks including Feed-forward back propagation, Generalized Regression and Radial Basis were implemented and were compared to evaluate their performance. Similar to earlier work with GA and Neural Nets, this work also established Feed-forward back propagation Neural Network as a preferred method for determining the design of airfoil since the technique presented better approximation results than other neural nets.

Keywords— Airfoil Optimization, Ant Colony Optimization, Bezier-PARSEC, C_p, Neural Network

I. INTRODUCTION

Airfoil design is one of the most challenging processes [1] in development of aircraft aerodynamic surfaces as it affects various aircraft performance parameters like lift, drag, spin-stall, cruise and turning radius [2]. Studies indicate that selecting the right design of airfoil with required characteristics reduces overall cost and improves the performance of air vehicle. Airfoil design largely depends on desire for high lift to drag ratio that is in conflict with the performance requirements [3].

There are two major techniques for designing an airfoil; direct and inverse [4]. First method involves designing a new or modifying an existing airfoil (UIUC Airfoil Database [5] and computing pressure distribution across the surface to achieve desired set of parameters. This approach may limit the approximation for desired specifications due to inherent limitations in airfoil's aerodynamics. For faster approximations, reduced degrees of freedoms are required but such reduction results in computational errors like round off, truncation and discretization error. In fact, determining the airfoil geometry should be based on requirements for aircraft's performance. Thus later method involves using desired operational characteristics and performance parameters unless the airfoil geometry so generated meets the desired criteria. To reduce the computational time and meet the required design criteria various techniques including CFD, fuzzy logic, neural networks [6] and heuristics based algorithms like PSO [7] and GA [8] have been implemented to advantage the aerodynamic design process.

This research, largely inspired by Saleem and Kharal [9], uses neural network based approach for airfoil generation exploiting Bezier-PARSEC 3434 parameterization rather than full coordinates for a given Cp. However, this research implements ACO to optimize Bezier-PARSEC unknown parameters instead of GA as in earlier work.

II. ARITIFICIAL NEURAL NETWORK

In machine learning and data mining, Artificial Neural Network is a set of learning algorithms modeled after neural network structure of the cerebral cortex and is used to approximate functions involving a larger number of the unknown input variables [10] Each neuron receives input from external sources or neighbors in the network, computes output and propagates to other neurons. Another function is the weight adjustments in the connections between neurons. Incremental learning is the technique by gathering information on cumulative error and consequently adjusting weight coefficients, w_{ij}. Mathematically, a Neural Network can be defined as a triple (*N*, *C*, *w*) where N is the set of neurons, C $\{(i, j)|i, j \in N\}$ is a set of connections, and function w((i, j)), shortened as w_{ij} is called weights between neurons i and j. For every neuron, there is an external input ϑ_j and an activation function F_j to establish the new activation level based on effective input of a neuron S_j and is determined by following propagation rule in "(1)".

$$S_j = \sum_i w_{ij}(t)\, y_i(t) + \theta_j(t) \tag{1}$$

ISBN: 978-1-61804-323-8

Besides, a threshold is also introduced as linear, non-linear or sigmoidal function [11] that helps avoid the situation when training is not successful at ||σ||>0. A threshold function for each neuron is given by "(2)"

$$F_{S_j} = \frac{1}{1 + e^{-S_j}} \tag{2}$$

A. Feed-forward Back propagation Neural Network

A feed forward Back Propagation Neural Network (FFBP) contains a multi layered interconnected feed forward structure where every layer gets input from below and gives output to layer above it. Back propagation is a learning technique where output values are compared to a desired value to calculate the error using a pre-determined error function. This value of error is then fed back through the network repeatedly for minimizing through neural network algorithm by adjusting weights for each network connection until the network converges to a bare minimum acceptable level of error [12] Generally, a non-linear optimization method called gradient descent is implemented where derivative of the error function is determined w.r.t. weights, that are adjusted till the reduction of error.

B. Radial Basis Function Neural Network

A Radial Basis Function Neural Network (RBF) consists of an input layer, a hidden layer with non-linear Radial Basis activation function and an output layer. For Radial Basis Neural Network, the input is modeled as vector of real numbers (R^n) while output is a scalar function φ, given in "(3)" by [13]

$$\varphi(x) = \sum_{i=1}^{n} a_i p(||x - c_i||) \tag{3}$$

where n is number of neurons, a_i is weight of neuron and c_i is center vector.

In Radial Basis Neural Networks, neurons respond to inputs close to their center in contrast with other neural networks. Although Radial Basis Neural Network requires more neurons for high dimensional input spaces, it can be trained faster than standard multi layered neural networks and have proven efficiency in regression and classification problems.

C. Generalized Regression Neural Network

A Generalized Regression Neural Network (GRNN) consists of one each input layer, pattern layer, summation layer and output layer. Training patterns are presented by neurons in pattern layer. In GRNN, pattern layer is connected to summation layer. Sum of weighted responses and un-weighted responses of pattern neurons are computed by two neurons in summation layers [9] The summation layer consists of both summation and single division units. Normalization of output is performed together both by summation and output layers. GRNN exhibit single pass learning algorithm with high parallel structure for estimating continuous variables and do not require iterative process as in multi-layered networks. GRNN converges to optimal regression even in noisy environments given a large number of sample data is available. Generalized Regression Neural Network is particularly advantageous with sparse data but as the training data increase, the error converges to zero.

III. PARSEC PARAMETERIZATION & BEZIER CURVES

PARSEC parameterization has the capability to describe the airfoil shape and its flow using engineering parameters [10] On the other hand, a Bezier curve is a parametric curve of degree n defined by polygon of n+1 vertex points called control points of nth order Bezier curve and is given by "(4)"

$$P(t) = \sum_{k=0}^{n} P_k \binom{n}{k} t^k (1-t)^{n-k} \tag{4}$$

where P_k is the kth control point while parameter t ranges from 0 to 1 with 0 at the zeroth control point and 1 at the nth control point. Eq. (5) gives Third order Bezier Curve

$$\begin{cases} x = x_a(1-t)^3 + 3x_b(1-t)^2 t + 3x_c(1-t)t^2 + x_d t^3 \\ y = y_a(1-t)^3 + 3y_b(1-t)^2 t + 3y(1-t)t^2 + y_d t^3 \end{cases} \tag{5}$$

Eq. (6) present fourth order Bezier Curve

$$\begin{cases} x = x_a(1-t)^4 + 4x_b(1-t)^3 t + 6x_c(1-t)^2 t^2 + 4x_d(1-t)t^3 \\ y = y_a(1-t)^4 + 4y_b(1-t)^3 t + 6y_c(1-t)^2 t^2 + 4y_d(1-t)t^3 \end{cases} \tag{6}$$

IV. BEZIER-PARSEC PARAMETERIZATION

Bezier-PARSEC parameterization is a technique in which Bezier Curves are described using PARSEC parameterizations [14] and is further subdivided into BP3333 and BP3434.

A. BP3333 Parameterization

BP3333 Parameterization employs third order Bezier Curves for camber shape and thickness of airfoil [15] Twelve PARSEC parameters represent Bezier control points as shown in Fig 1.

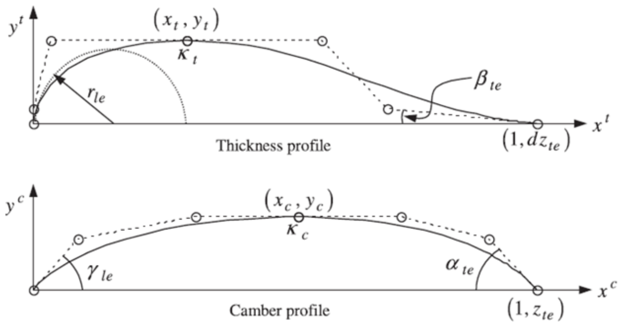


Fig 1 : BP3333 Bezier PARSEC Control Points and Respective Airfoil Geometry

Main advantages of BP3333 include close relevance to airfoil aerodynamics parameters, faster optimization, continuity characteristics, reduced deviation of design process and avoidance of sharp leading edges. Disadvantage of this technique is reduced degree of freedom resulting in failure to parameterize airfoils having radical camber trailing edge

B. BP3434 Parameterization

BP3434 Parameterization depends on 10 PARSEC parameters and 5 Bezier parameters for airfoil shape representation. Here, camber and thickness leading edge of airfoil is defined by third order Bezier Curves while fourth order Bezier Curves are used to define camber and thickness trailing edge of airfoil shape [15] This allows increased degree

ISBN: 978-1-61804-323-8

of freedom for trailing edge parameterization of airfoil as shown in Fig 2.

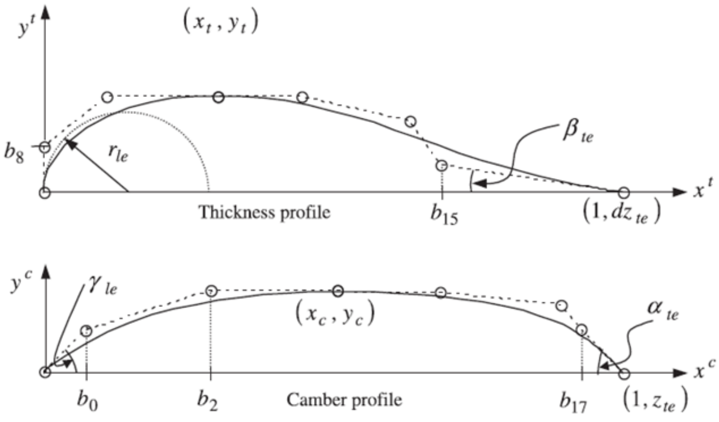


Fig 2 : BP3434 Bezier PARSEC Control Points and Respective Airfoil Geometry

BP3434 proves to be efficient than BP3333 when airfoil camber becomes negative along any part of the chord length. However, the convergence speed for this method reduces due to greater number of variables as compared to BP3333. In presence of high computing numerical computers, the convergence speed of BP3434 can be compensated for effective application of the method.

V. ANT COLONY OPTIMIZATION

Ant Colony Optimization (ACO) is the meta-heuristic motivated from the working of natural ants that suggests that ants follow different paths to reach food source initially. Thus the ants with shortest path would reach the source in least time than the longer paths [16] Ant Colony Optimization is an algorithm where artificial ants are used to probabilistically construct solutions guided by higher pheromone trails and promising heuristic information [17] In actual, ants implement a randomized construction heuristics that differ from greedy heuristics by adding a probabilistic component to partial solution than a deterministic one. Generally, ACO algorithm consists of two phases. In first phase, artificial ants construct a solution where in second phase, pheromone trail is updated by first reducing by an evaporation factor to avoid unlimited accumulation followed by adding pheromone proportionate to quality of their solutions [18]. Thus most important is to update pheromone for generating quality solutions in future iterations of algorithm. ACO algorithms can be considered as competitive solution technique where previous solutions known to be part of good solutions are used to generate even better solutions in future cycles [19].

VI. METHODOLOGY

In this research work, our methodology was quite similar to earlier work; however, ACO was preferred as a choice for optimization technique instead of GA to determine unknown Bezier-PARSEC parameters.

A. Airfoil Representation

A vector of 71 points is used to represent x-y coordinates of an airfoil where x_i ranges from 1 to 0 for upper airfoil curve and lower airfoil curve, thus only values for y change which determine the shape of both curves.

Mean Camber Line is a line at equal distance from both upper and lower surfaces of airfoil. Therefore, camber curve y points were obtained by taking average of upper and lower coordinates corresponding to the same x coordinate. These upper and lower coordinates were divided by chord length for non-dimensionalizing. The camber profile of an airfoil is calculated by "(7)", "(8)", "(9)" and "(10)"

$c-|x_1-x_{36}|$ for i=1 to 36 and j=36 to 71

$$y_i^u = \frac{y_i}{c} \quad \text{and} \quad y_j^l = \frac{y_j}{c} \qquad (7) \,\&\, (8)$$

$$x_i^c = \frac{x_i}{c} \quad \text{and} \quad y_i^c = \frac{y_i^u + y_j^l}{2} \qquad (9) \,\&\, (10)$$

Thickness curve used to define the airfoil thickness is the difference between the camber curve and upper curve of the airfoil i.e.

$$y_i^t = y_i^u - y_i^c \qquad (11)$$

Next a two dimensional analysis of airfoil was carried out using Panel Method to obtain values for lift coefficient C_l quarter-chord pitching moment coefficient C_m and drag coefficient C_d at ten angles of attack α. Thus the airfoil would be represented by x_i^c, y_i^c, C_l, C_d, C_m and α.

B. Calculating Bezier-PARSEC Parameterization

Table 1 presents the required parameters for Bezier-PARSEC

Table I : Known Bezier-PARSEC Parameters

Parameters	Caculations
Maximum Thickness Point	$y_t = C^t(\{x_t \mid \frac{dC^t}{dx_i}\Big\vert_{x_i=x_t} = 0\})$
Maximum Camber Point	$y_c = C^c(\{x_c \mid \frac{dC^c}{dx_i}\Big\vert_{x_i=x_c} = 0\})$
Trailing Edge Vertical Displacement	$Z_{te} = C^c(x)\vert_{x=1}$ and $dZ_{te} = C^t(x)\vert_{x=1}$
Trailing Camber Line Angle	$\alpha_{te} = -tan^{-1}(\frac{dC^c}{dx}\vert_{x=1})$
Trailing Wedge Angle	$\beta_{te} = -tan^{-1}(\frac{dC^t}{dx}\vert_{x=1})$
Leading Edge Direction	$\beta_{te} = -tan^{-1}(\frac{dC^t}{dx}\vert_{x=1})$
Leading Edge Radius	r_{le}

While ten parameters are calculated using Bezier-PARSEC equations, there is no specific mathematical expression for finding remaining five parameters i.e., b_0, b_2, b_8, b_{15} and b_{17} and therefore are calculated by curve fitting. Since actual airfoil is known, Bezier Curves with correct five control points would suffice given a smallest Sum-of-Least-Square Error.

Table II shows the four curves and corresponding unknown Bezier points.

Table II : Unknown Bezier-PARSEC Parameters

Curve	Bezier Curve	Order	Unknown Bezier Control Points
Camber	Leading Edge	3^{rd}	b_0, b_2
	Trailing Edge	4^{th}	b_{17}
Thickness	Leading Edge	3^{rd}	b_8
	Trailing Edge	4^{th}	b_8, b_{15}

C. Optimization of Unknown Bezier Control Points Using ACO

To determine optimal value of these unknown parameters, Ant Colony Optimization was implemented requiring fitness functions for each Bezier Curve that was equal to the difference between Bezier generated and actual airfoil. For this a Simple ACO code was written to determine each of these parameters i.e., b_0, b_2, b_8, b_{15} and $b_{17.}$ In ACO, 6 ants were used to determine the optimal path to the destination and since the destination point was unknown; therefore, SSE for each curve was calculated for each generated point. Thus, a decrease in SSE over the path indicates that the ant is close to the destination point and vice versa. The pheromone is inversely proportional to the distance so the path with least distance or least SSE would have maximum pheromone. For each value of b_0, a corresponding value of b_2 is calculated through ACO. Thus a number of combinations (pair of b_0 and b_2 values) are made where pair with the least SSE is finally chosen. Same approach was used for b_8 and b_{15} while value of b_{17} was calculated separately. Fig 3 present flow charts for the method used.

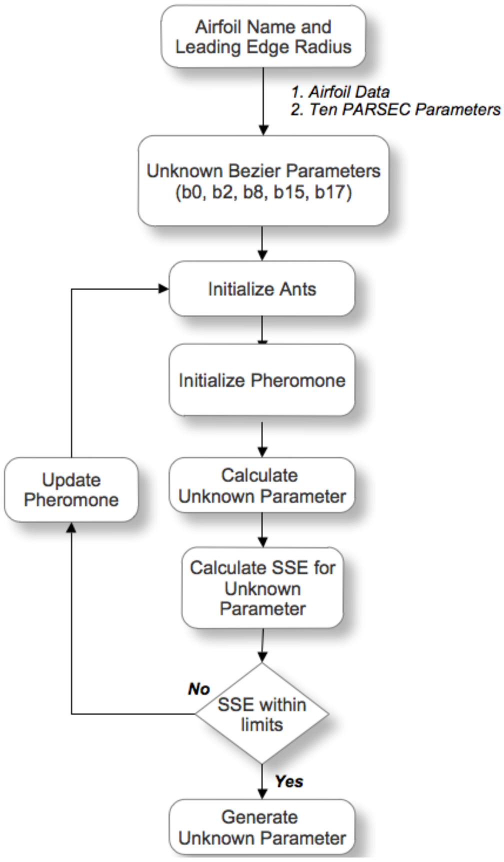


Fig 3 : Ant Colony Optimization Methodology

D. Error Calculation

All 15 BP3434 parameters determined are used for generation of airfoil geometry. The error is calculated by comparing Bezier generated airfoil with actual airfoil. To calculate this error, at a certain x-value, y-value from parameterized and actual airfoils should relate to this x-value. The main challenge was to determine y-values of Bezier parameterized airfoil corresponding to these x-values. After generating x and y values of trailing and leading edge of thickness curve, these are arranged into a single set of x-y array in which first element corresponds to leading edge followed by trailing edge. Then cubic spline interpolation is used to fit a curve in the vector of x and y values which is then evaluated for 36 x-values of actual airfoil. Same procedure was followed for camber curve. These thickness and camber curves can be used to determine the shape of airfoil. The airfoil geometries of parameterized and actual airfoils are then plotted against same axis for comparison.

Fig 4 shows flow chart for SSE calculations while Fig 5 presents results for Eppler 433 sailplane parameterized airfoil.

E. Neural Networks Estimations

Neural Networks of three types as discussed in Section 2 were implemented in this research work. A 10X4 matrix of Cl, Cd, Cm at ten angles of attack for 500 heterogeneous airfoils was input to neural network while target was 15 Bezier-PARSEC parameters for airfoil generation.

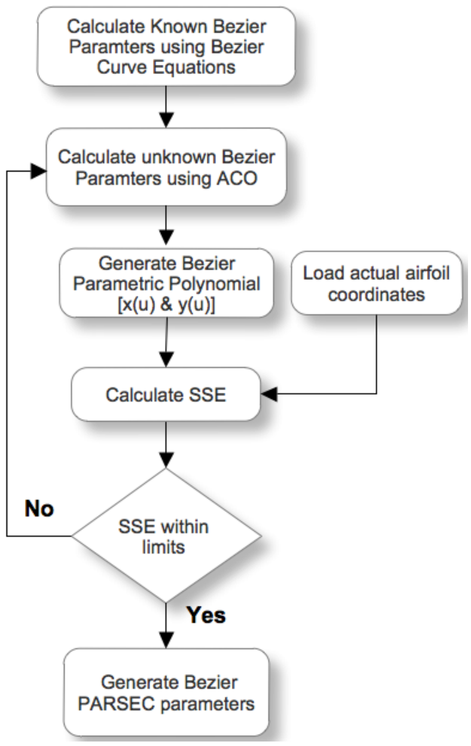


Fig 4 : SSE Calculations for Bezier-PARSEC Parameters

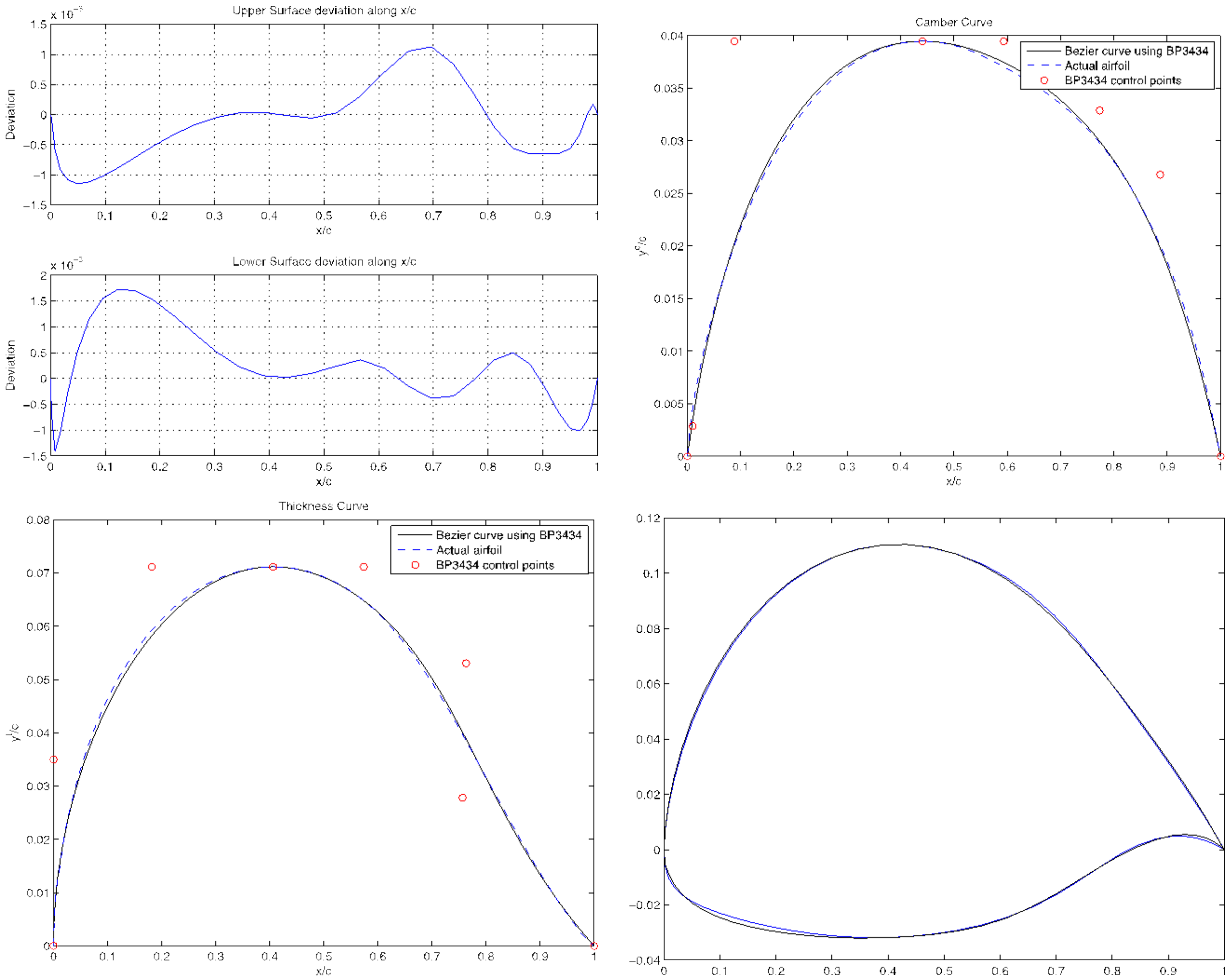


Fig 5 : Bezier Parameterization Results for Eppler 433 Sailplane Airfoil

ISBN: 978-1-61804-323-8

VII. RESULTS AND DISCUSSIONS

A. Comparison of ACO Results with GA

Implementation of ACO for finding unknown Bezier Curve parameters proved to be more efficient than Genetic Algorithm. We were able to achieve a precision level of $\leq 1X10^{-5}$ as compared to GA based version of the program. Also time to optimize the missing BP3434 parameters was greatly reduced. For example, Eppler 433 Sailplane airfoil took 30.905144 seconds to optimize BP3434 missing parameters using ACO as compared to GA that took 87.869966 seconds for optimization of said airfoil using 2.7GHz Processor and 4GB RAM. Table III gives a comparison of ACO and GA optimizations for few airfoils for reference.

Table III : Comparison of ACO and GA Optimization Results

Airfoil	Ant Colony Optimization		Genetic Algorithm	
	Time (Seconds)	Level	Time (Seconds)	Level
Eppler E433	30.905144	$\leq 1X10^{-5}$	87.869966	$\leq 1X10^{-4}$
NACA 65(4)-421	55.187357	$\leq 1X10^{-5}$	90.952194	$>1X10^{-4}$
Eppler E335	65.389595	$\leq 1X10^{-5}$	109.694796	$>1X10^{-4}$
Gottingen GOE426	44.489090	$\leq 1X10^{-5}$	82.259980	$\leq 1X10^{-4}$
Eppler E399	55.089536	$\leq 1X10^{-5}$	94.445729	$\leq 1X10^{-4}$

From Table III, we see that optimization time has remarkably been reduced to almost half for above airfoils.

B. Results of Neural Networks

As discussed above, three types of neural networks were implemented and tested against 500 airfoils for training and 200 airfoils unknown to the neural nets. Consolidated results for these airfoils is shown in Table IV.

The results from Table IV show that Feed Forward and Back Propagation has proved to be more promising in terms of better performance as indicated by increased fraction of both known and unknown airfoils within acceptable MSE values. On the other hand, GRNN and RBF showed improved efficiency with known airfoils than for the unknown airfoils. Comparison of Results for a known to network airfoil (Eppler 399 airfoil) and an unknown to network airfoil (Gottingen 426 airfoil) to the three types of neural networks is shown in Fig 6

The plots for Gottingen 426 airfoil and Eppler 399 airfoil support application of Feed Forward Back Propagation Neural Network for solving this problem. However, results from RBF and GRNN largely favour known to network airfoils than unknown airfoils as is evident from RBF and GRNN plots for Gottingen 426 airfoil. Results for 200 airfoils unknown to network also support similar findings. MSE for GRNN and RBF is higher than FFBP with RBF performing the worst with a high MSE.

Table IV : Comparison of Test Results for Three Neural Nets

Artificial Neural Network	$\leq 1X10^{-5}$		$\geq 1X10^{-5}$	
	Count	%age	Count	%age
Feed Forward and Back Propagation				
Known Airfoils (500)	273	54.6	227	45.4
Unknown Airfoils (200)	113	56.5	87	43.4
Radial Basis Neural Network				
Known Airfoils (500)	394	78.8	106	21.2
Unknown Airfoils (200)	47	23.5	153	76.5
Generalized Regression Neural Network				
Known Airfoils (500)	363	72.6	137	27.4
Unknown Airfoils (200)	78	39.0	122	61.0

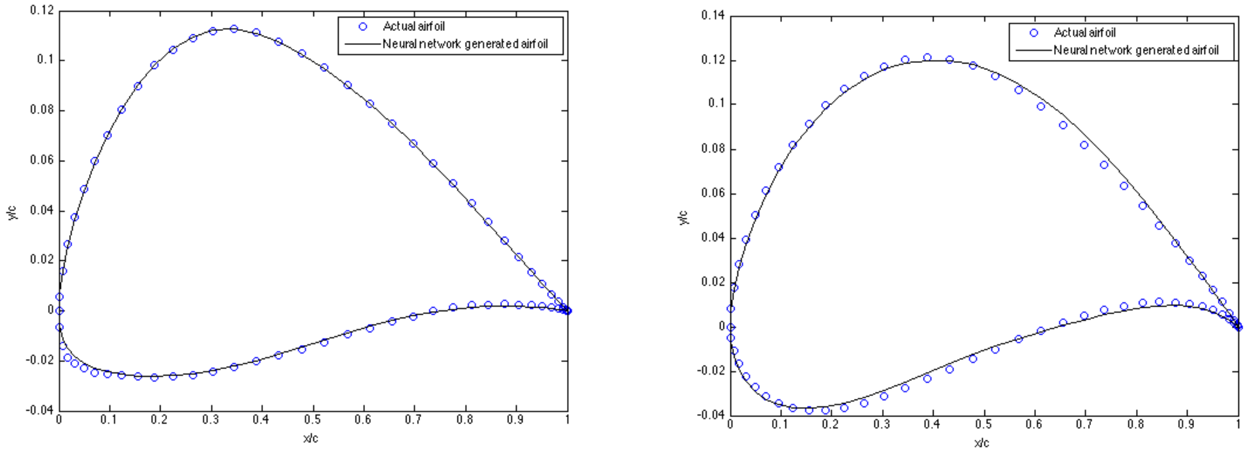
C. Regression Analysis

A post training regression analysis was performed to analyze the neural networks. In this analysis, the output of neural networks for known targets was compared. Thus neural network output would match the target values and would ideally be a straight line with 45° slope passing through the origin as shown in Fig 7.

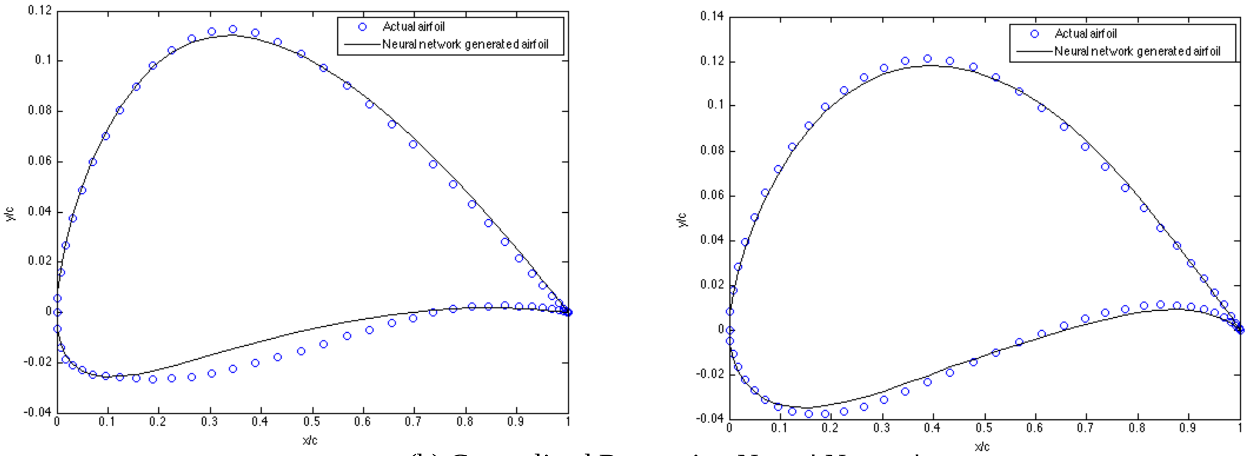
Fig 7 shows that performance of Feed Forward and Back Propagation is better than other two types of neural nets as indicated by the high regression values and low training R-values. On the other hand, both Generalized Regression Neural Network and Radial Basis Neural Network have higher R-values but shown poor results with test and validation data. The main reason is their architecture as both determine distance between input and weight vectors, which are incrementally multiplied by biased vectors. This would lead an input close to weight vector, produce an output close to unity while output would be close to zero if input is different from weight vector.

VIII. CONCLUSION

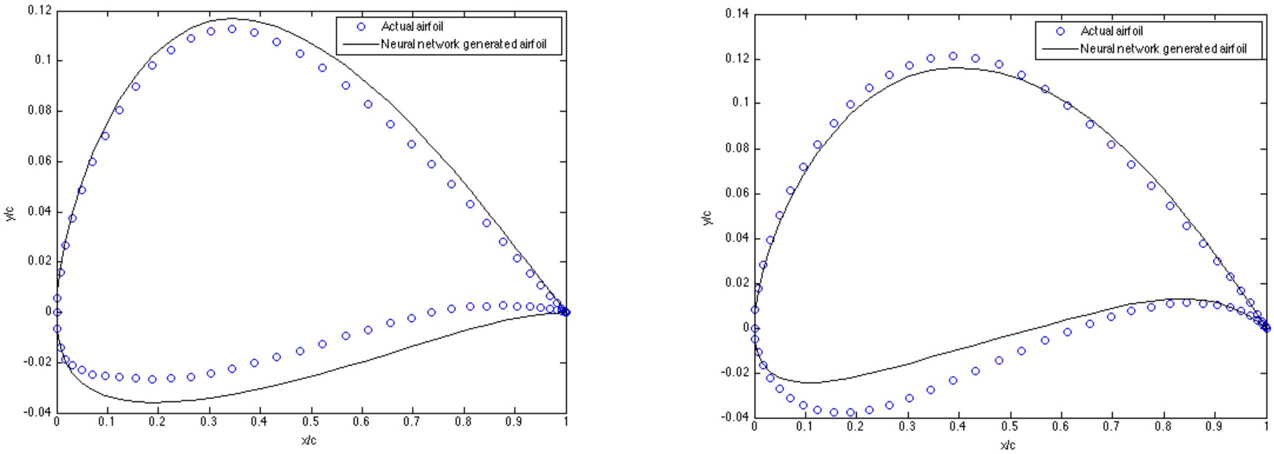
This work determines airfoil geometry for a given C_p using Neural Network and Bezier-PARSEC parameters. The main consideration of this paper is to use Ant Colony Optimization technique to optimize missing BP3434 parameters instead of complete set of airfoil coordinates. Further, three types of Neural Networks; Feed Forward and Back Propagation, Radial Basis and Generalized Regression were employed. Similar to earlier findings with GA based code, we proved that Feed-forward and Back Propagation exhibited greater efficiency than the other two types of Neural Networks. However, we were able to achieve higher precision with reduced time for optimization using ACO to determine missing BP3434 parameters. Besides, percentage of known and unknown airfoils with precision $\leq 1X10^{-5}$ has shown a slight increase.



(a) Feed Forward and Back Propagation Neural Network



(b) Generalized Regression Neural Network

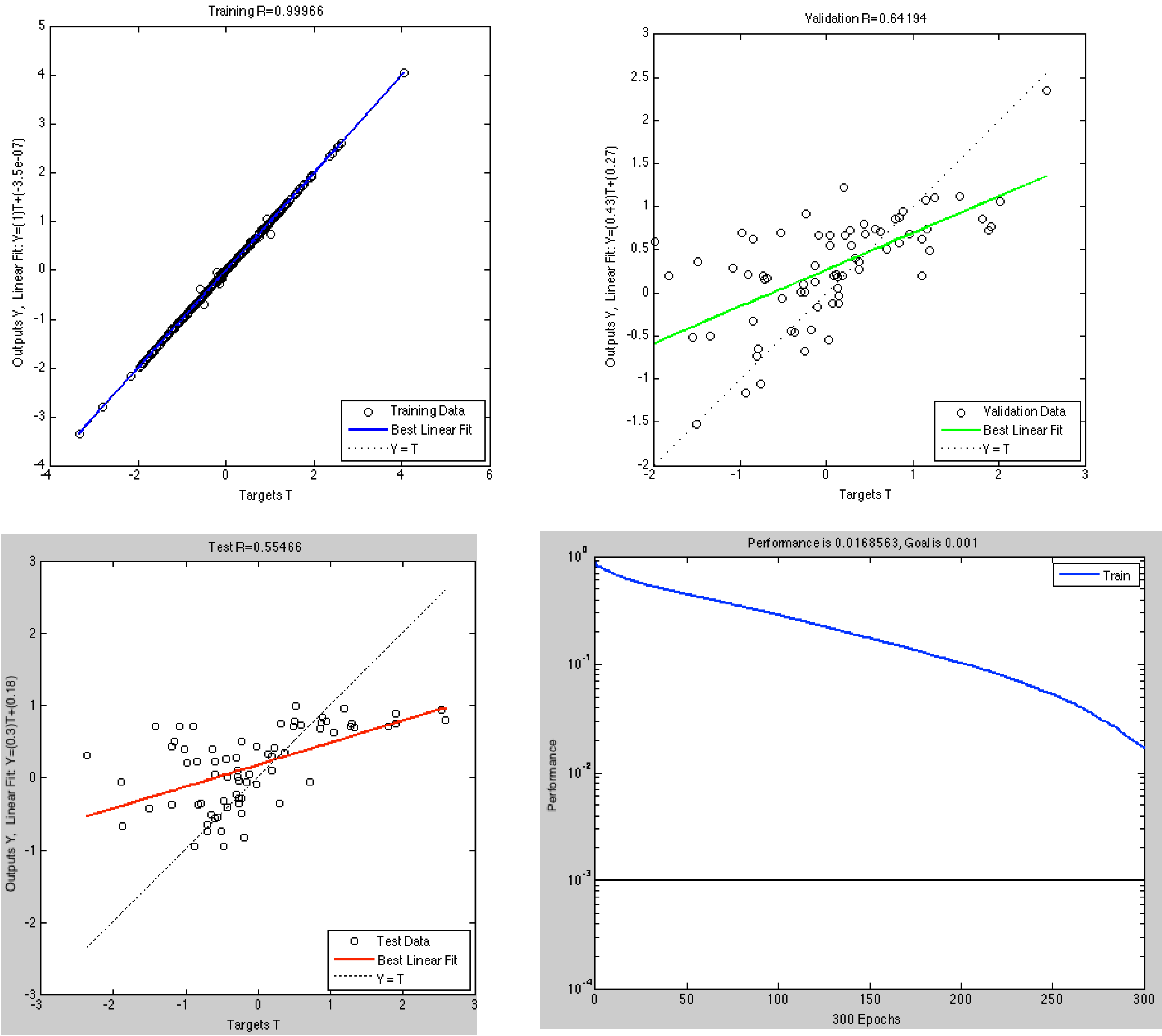


(c) Radial Basis Neural Network

Gottingen GOE426 Airfoil Eppler E399 Airfoil

Fig 6 : Comparison of Results of Known to Unknown Airfoil to Neural Network

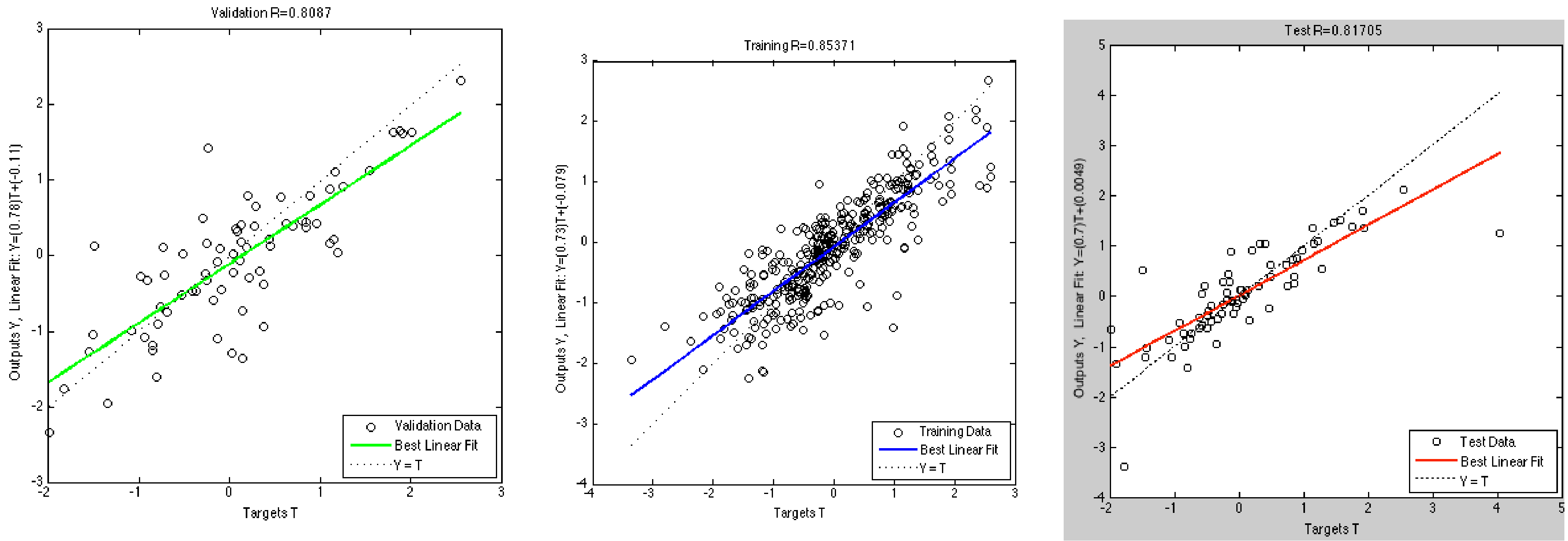
ISBN: 978-1-61804-323-8



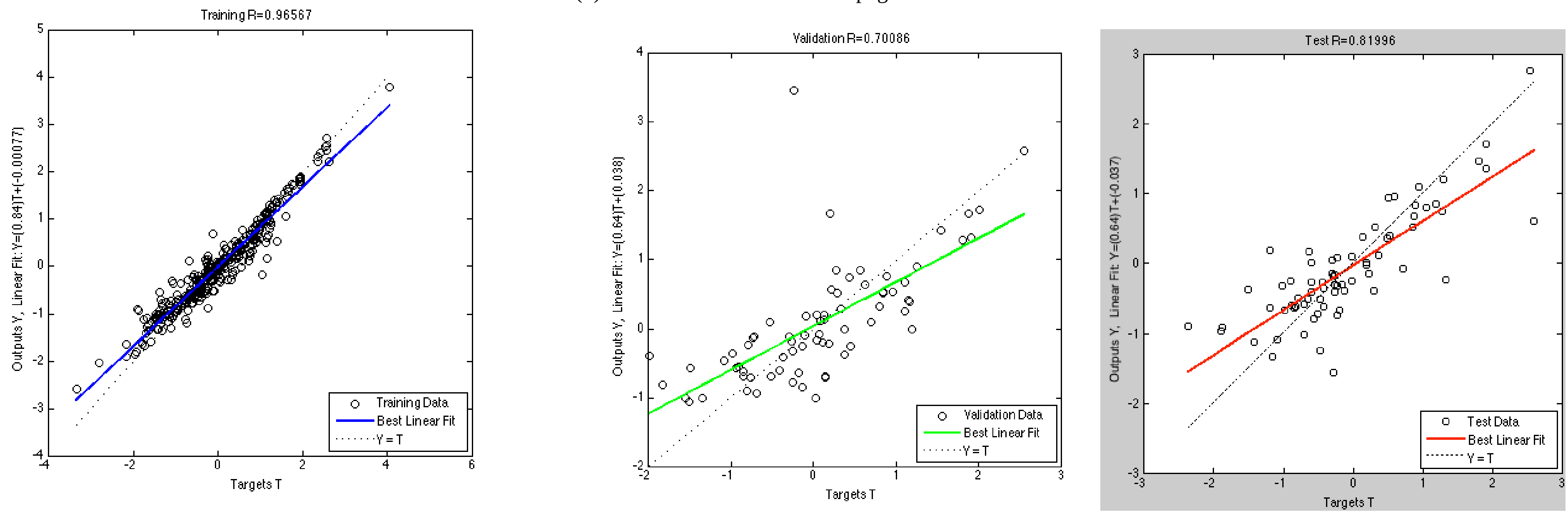
(a) Radial Basis Neural Network

IX. FUTURE WORK

We have implemented Simple ACO in this research work. Future works may consider implementation of other extensions of ACO techniques like Elitist AS, Ant-Q, Max-Min As, Hyper-cube AS and etc to achieve high performance in order to further reduce the optimization level and attain higher level of precision.



(b) Feed Forward and Back Propagation Neural Network



(c) Generalized Regression Neural Network

Fig 7 : Regression Analysis

REFERENCES

[1] G. Trapani, T. Kipouros and M. Savill, 'Computational Aerodynamic Design for 2D High-Lift Airfoil Configurations', 2010.

[2] K. Lane, 'Novel Inverse Airfoil Design Utilising Parametric Equations', Graduate, California Polytechnic State University, 2010.

[3] Z. Zhu, F. Hongyan, Y. Rixin and L. Jie, 'Multi-objective Optimization Design of Airfoil and Wing', *Science in China Series E Technological Sciences*, vol. 47, no. 1, pp. 15-25, 2004.

[4] P. Hewitt and S. Marques, 'Aerofoil Optimisation Using CST Parameterisation in SU2', in *Royal Aeronautical Society Applied Aerodynamics Group Conference*, Bristol, 2014.

[5] M-selig.ae.illinois.edu, 'UIUC Airfoil Data Site', 2015. [Online]. Available: http://m-selig.ae.illinois.edu/ads/coord_database.html. [Accessed: 11- Jan- 2014].

[6] A. Hacioglu, 'Fast Evolutionary Algorithm for Airfoil Design via Neural Network', *AIAA Journal*, vol. 45, no. 9, pp. 2196-2203, 2007.

[7] U. Wickramasinghe, R. Carrese and X. Li, 'Designing Airfoils Using a Reference Point Based Evolutionary Many-objective Particle Swarm Optimization Algorithm', *IEEE Congress on Evolutionary Computation*, 2010.

[8] M. Ebrahimi and A. Jahangirian, 'Aerodynamic Optimization of Airfoils Using Adaptive Parameterization and Genetic Algorithm', *Journal of Optimization Theory and Applications*, vol. 162, no. 1, pp. 257-271, 2013.

[9] A. Kharal and A. Saleem, 'Neural Networks Based Airfoil Generation for a Given Cp Using Bezierâ€“PARSEC Parameterization', *Aerospace Science and Technology*, vol. 23, no. 1, pp. 330-344, 2012.

[10] G. Sun, Y. Sun and S. Wang, 'Artificial neural network based inverse design: Airfoils and wings', *Aerospace Science and Technology*, vol. 42, pp. 415-428, 2015.

[11] K. Thinakaran and R. Rajasekar, 'Design of Airfoil Using Inverse Procedure and Neural Network', *An International Journal of Advanced Computer Technology*, vol. 2, no. 9, pp. 285-290, 2013.

[12] L. Prabhu and J. Srinivas, 'Flutter Prediction Using Neural Networks-based Hybrid Optimization Scheme', *International Journal of Research in Aeronautical and Mechanical Engineering*, vol. 2, no. 4, 2014.

[13] M. Ahmed and N. Qin, 'Surrogate-Based Aerodynamic Design Optimization: Use of Surrogates in Aerodynamic Design Optimization', in *13th Conference on Aerospace Sciences & Aviation Technology*, Cairo, 2009.

[14] R. Derksen and T. Rogalsky, 'Bezier-PARSEC: An Optimized Aerofoil Parameterization for Design', *Advances in Engineering Software*, vol. 41, no. 7-8, pp. 923-930, 2010.

[15] N. Salunke, R. Juned and S. Channiwala, 'Airfoil Parameterization Techniques: A Review', *American Journal of Mechanical Engineering*, vol. 2, no. 4, pp. 99-102, 2014.

[16] M. Darigo and L. Gambardella, 'Ant Colonies for the Travelling Salesman Problem', *BioSystems*, vol. 43, pp. 73-81, 1997.

[17] M. Darigo and T. Stutzle, *Ant Colony Optimization*. Cambridge: The MIT Press, 2004.

[18] J. Bell and P. McMullen, 'Ant Colony Optimization techniques for The Vehicle Routing Problem', *Advanced Engineering Informatics*, vol. 18, no. 1, pp. 41-48, 2004.

[19] G. Fainekos and K. Giannakoglou, 'Inverse Design of Airfoils Based on a Novel Formulation of the Ant Colony Optimization Method', *Inverse Problems in Engineering*, vol. 11, no. 1, pp. 21-38, 2003.

ISBN: 978-1-61804-323-8