Applied Mathematical Modelling 36 (2012) 4898–4907

Contents lists available at SciVerse ScienceDirect

Applied Mathematical Modelling

journal homepage: www.elsevier.com/locate/apm

An airfoil optimization technique for wind turbines

A.F.P. Ribeiro *, A.M. Awruch, H.M. Gomes

Graduate Program in Mechanical Engineering, Federal University of Rio Grande do Sul, Brazil

ARTICLE INFO

Article history:
Received 25 April 2011
Received in revised form 26 November 2011
Accepted 4 December 2011
Available online 30 December 2011

Keywords:
Computational fluid dynamics
Genetic algorithms
Artificial neural networks

ABSTRACT

Optimization algorithms coupled with computational fluid dynamics are used for wind turbines airfoils design. This differs from the traditional aerospace design process since the lift-to-drag ratio is the most important parameter and the angle of attack is large. Computational fluid dynamics simulations are performed with the incompressible Reynolds-averaged Navier–Stokes equations in steady state using a one equation turbulence model. A detailed validation of the simulations is presented and a computational domain larger than suggested in literature is shown to be necessary. Different approaches to parallelization of the computational code are addressed. Single and multiobjective genetic algorithms are employed and artificial neural networks are used as a surrogate model. The use of artificial neural networks is shown to reduce computational time by almost 50%.

1. Introduction

With the increasing importance of renewable energy, airfoil design, specifically for wind turbines, has become a fundamental issue. Although airfoil optimization based on genetic algorithms has been performed by several different research groups in the past decade [1–4], most of the available literature is applied to aircraft design. Optimization processes of entire airplanes have been carried out using computational fluid dynamics (CFD) and different types of algorithms [5,6], while wind turbines are commonly optimized using blade element theory, changing the chord and twist angle of the blade section [7], without changing the shape of the airfoil or considering three-dimensional modifications, such as swept blades or tip devices.

Airfoils created for airplanes are occasionally used in wind turbines [8], even though the design criteria are not the same for both cases. For instance, in wind turbines the airfoil can be used at a higher angle of attack, sometimes even achieving stall, and the most important parameter is the lift-to-drag ratio [9]. In aircraft design, the objective of the design process might be quite different, such as decreasing the drag for a fixed lift coefficient [10]. Another concern is that Reynolds and Mach numbers are much lower for wind turbines in common conditions than they are for airplanes, which may change the flow considerably.

Wind turbine optimization is usually performed with blade element theory. This takes two-dimensional information of the employed airfoils and the blade design is performed taking into account momentum conservation along a stream tube that crosses the rotor [9,11]. This type of calculation is very quick, so it has been used extensively in blade optimization, even including aeroelastic analysis [7,12,13]. However, such methods have many limitations, since viscous and three-dimensional effects are not taken into account. Hence, results found with these well established methods can be quite different from reality [14]. Although three-dimensional optimization coupled with CFD has been performed for turbomachinery [15], to the authors' knowledge, this has not been applied to wind turbines. More details of the state of the art can be found in [16].

[*] Corresponding author. Address: Av. Osvaldo Aranha, 99, Porto Alegre, RS, Brazil. Tel.: +55 51 9696 3624.
E-mail address: ribribeiro@yahoo.com.br (A.F.P. Ribeiro).

0307-904X/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.apm.2011.12.026

In aircraft design, the optimization process can typically be divided into four different problems: geometry generation, numerical simulation, application of the optimization algorithm and the surrogate model. Geometry generation is related to the method used to translate the surface of the airfoil into numbers, which can then be used as variables in the optimization process. Several methods are available to do this, ranging from formulations designed specifically for airfoils, such as the PARSEC method [4], to parametric curves, which have a more general purpose. Examples of parametric curves routinely employed in airfoil optimization are splines [17], B-splines [18,19] and Bézier curves [3,20–24]. Other methods are occasionally used by authors, such as the summation of analytical functions of the Hicks and Henne method [25,26], quadratic equations [27] or more complex combinations, like the combination of ellipses and splines [28].

In the last decade, studies about aerodynamic simulation have focused on CFD instead of less robust methods. When the Reynolds number is high enough for viscous effects to be negligible, the compressible Euler equations are used [17,24,27]. Otherwise, the compressible or incompressible Reynolds Averaged Navier–Stokes equations are employed. These equations are coupled with turbulence models with various degrees of complexity. Zero equations models, such as the Baldwin–Lomax, are still utilized by some groups [3], while the one equation Spalart–Allmaras turbulence model has also been quite popular [5,23,26]. Two equations models, such as the standard $k - \varepsilon$ model are commonly used [4,22,29], but more modern models, such as the $k - \omega$ SST [19] have also been employed .

The optimization algorithm usually falls within two categories: gradient based methods and heuristic algorithms. Gradient based methods have been successfully applied to the optimization of turbomachinery airfoils [28], multiple component wing sections [2] and even to entire airplanes [6]. These methods are very popular due to their speed. However, in practice, they seldom converge to global optima. Hence, it becomes necessary to check various initial conditions in order to have some certainty in the design. In this sense, these methods are not robust [30]. Although considered as being slower, genetic algorithms, which are the most popular of the heuristic algorithms, are more reliable and robust, even though convergence is not proven. They have been applied to several different cases, such as airfoils [4,21,22], wings [3,5,18,24], and also entire airplanes [31]. Hybrid methods, combining these two categories of algorithms, are sometimes employed [19,32,33].

A great number of surrogate models are routinely applied to aerodynamic optimization. Artificial neural networks have been used for airfoils [21,28], multiple component wing sections [1], and three dimensional optimization of the wing tip [19]. Although quite popular, polynomial response surfaces have the disadvantage of requiring statistical treatment to obtain the appropriate design of experiments, and degree of the polynomial based on the available data. For unknown data this implies prior investigation in order to accurately fit data without using unnecessary polynomial terms. Artificial neural networks has the advantages of being easy to use, behaving well even with unknown data, and are equivalent to other methods for nonlinear regression, such as Universal Kriging [34].

This work is dedicated to the study of optimization methods in order to create an efficient technique to design airfoils for wind turbines, using an incompressible and relatively low Reynolds flow with high angles of attack. A surrogate model is also used, looking to decrease computational time. Despite of its simplicity, the current approach indicates a methodology that can be used in a more sophisticated way, dealing with three-dimensional modeling and accounting for other fluid effects on the blade. Results presented here may be used as a starting point for a more realistic blade shape optimization, which is at this moment an ongoing research. Section 2 presents a brief description of the methods used in this work. In Section 3 the numerical simulations are validated with experimental results and a comprehensive grid study is presented. In Section 4 results are shown. Conclusions are presented in Section 5.

2. Methods used in the aerodynamic optimization

The optimization procedure cycle is shown in Fig. 1, displaying the involved steps involved in the process. If a surrogate model is not employed, the geometrical variables are given by the optimization algorithm and they are used to generate the geometry of the airfoil, which is then utilized in a numerical simulation in order to obtain the objective function, which is returned to the optimization algorithm. On the other hand, if a surrogate model is used, the variables are turned into the objective function directly by means of artificial neural networks, which drastically reduces the computational time. The methods used to generate the geometry, to calculate the aerodynamic performance, and to perform the optimization, as well as the employed surrogate model are presented in the following sections.

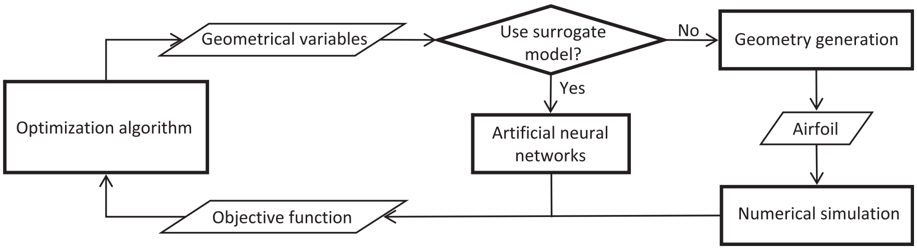

Fig. 1. Flow chart of the optimization cycle employed in this work.

2.1. Geometry generation

An important part of the optimization process is the parameterization of the geometry, since it allows the translation of the variables generated by the algorithm into the actual airfoil. According to [1], a parameterization method for aerodynamic optimization must be:

(1) Flexible, in order to cover a large search space, allowing "non-traditional" shapes to show up.
(2) Able to keep the number of design parameters as low as possible.
(3) Free of discontinuities.
(4) Free of variables that affect negligibly the aerodynamic performance, which would delay convergence.
(5) Accomplished so that its design variables are directly linked to the constraints.

Bézier curves are used in this work, since they are widely accepted in aerodynamic optimization [3,20–22,24] and satisfy the conditions previously established, with the exception of the last one. They are parametric curves given by the following equation:

$$B(t) = \sum_{i=0}^{n} \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i} P_i, \tag{1}$$

where n is the degree of the curve (which has $n + 1$ points) and P_i are the points defining the curve. Fig. 2 shows a GA(W)-1 airfoil [10] with chord c generated using two Bézier curves, one for the extrados and the other one for the intrados, each one with 6 points. The corresponding points used to generate both curves are also illustrated.

2.2. Numerical simulation

Since the Mach number used in wind turbines is low, the incompressible steady state Reynolds-averaged Navier–Stokes equations are used. The Reynolds stress tensor is modeled with the Spalart–Allmaras turbulence model [35], as it is often employed in aerodynamic optimization of airfoils and airplanes [5,23,27]. Numerical simulations are performed with the finite volume, cell-centered commercial code Fluent. The pressure-velocity coupling is accomplished with the Semi Implicit Linked Equations (SIMPLE [36]) algorithm. Central differences are employed to interpolate the pressure on cell faces and the Quadratic Upwind Interpolation for Convective Kinematics (QUICK [37]) scheme is used to interpolate the convective terms of the other variables.

2.3. Genetic algorithms

According to [33], some of the important characteristics of optimization algorithms are:

(1) Generality of the formulation.
(2) Robustness, in the sense of avoiding local optima, looking for global optimal points.
(3) Capability of handling multiple objectives.
(4) Computational efficiency.

Genetic algorithms [38] are based on Darwin's evolution theory, with an initial randomly generated population where the fittest individuals have higher chances of reproducing, with operations such as crossover, mutation, and elitism involved in

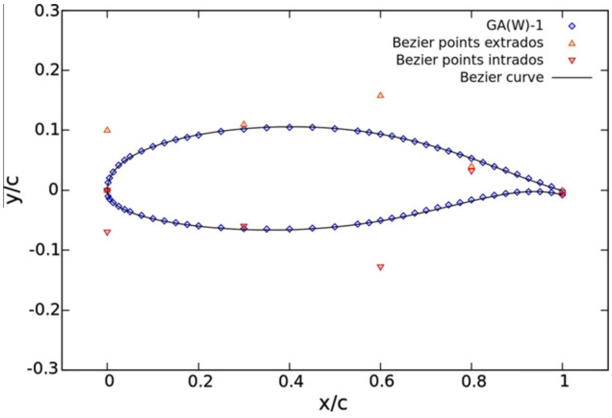

Fig. 2. GA(W)-1 airfoil approximated with Bézier curves.

this process. These algorithms are used to solve many different types of problems and are very robust, as they tend to search for global optima. Additionally, they have the ability to handle multiple objectives. The disadvantage of this method is that it is not very efficient, often requiring more computational time than other algorithms, especially gradient-based methods.

In the present study, the open source software NSGA-II [39] is used for the optimization process. The crossover and mutation probabilities are set to 90% and 5%, respectively. The distribution index for crossover and mutation are 12 and 20, respectively.

2.4. Artificial neural networks

An artificial neural network (ANN) is a parallel distributed processor composed by simple processing units (neurons) with a natural propensity for storing knowledge and making it available for use. It resembles the human brain, since the knowledge is acquired through a learning process and the interneuron connection strengths (synaptic weights) are used to store the obtained knowledge [40]. The ANN library used in this work is FANN (Fast Artificial Neural Networks). It is an open source project started by Nissen [41]. A multilayer perceptron ANN was employed with three layers: an input layer, composed by the input variables and a bias; a hidden layer, composed by $2m + 1$ (where m is the number of inputs) neurons and a bias; and an output layer. The activation function used in the output layer is linear and in the hidden layer a hyperbolic tangent activation function is employed.

An ANN was used here in the following way: firstly, the GA runs by itself, without any surrogate models. After a certain number of generations, the optimization process stops and all of the CFD results are used to train the ANN. The training is performed using a back propagation algorithm. After that, the GA is fully coupled with the ANN and the CFD coupling is disabled. Hence, all of the subsequent results are calculated using the ANN, which takes a negligible amount of time, when compared to the first part of the process, where CFD is used.

3. Validation of the numerical simulations

To guarantee the reliability of the numerical simulations performed in this work, a validation process is carried out. The airfoil used for the validation is the GA(W)-1 [10], due to the fact that it has detailed experimental results for relatively low speeds and high angles of attack (α). The numerical results are compared with the experimental ones for Reynolds and Mach numbers equal to $6e + 6$ and 0.15, respectively, and with trip wires at $0.08c$ to generate a turbulent boundary layer.

3.1. Grid independence

The validation process begins with the generation of three 2D grids: a coarse (C), a medium (M), and a fine (F) mesh. These grids are created following the recommendations of [42], with the following characteristics:

(1) In the boundary layer region, $y+ \leqslant [1,2/3,4/9][C,M,F]$, with the second layer having the same height as the first one, followed by a growth rate of 1.2.
(2) The far field is located at about $100c$ of the obstacle.
(3) A grid spacing of 0.1% of the chord at the leading and trailing edges.
(4) A number of cells at the blunt trailing edge of [8,12,16] $[C,M,F]$.
(5) A refinement ratio of 1.5 in each direction for the different grids.

A fourth, extra fine (XF) grid is also suggested by Mavriplis et al. [42], but in the present case this will be shown to be unnecessary. The high quality structured O-grid used is shown in Fig. 3. The far field is divided into two regions, so that the inlet can have a prescribed velocity boundary condition, while pressure is prescribed at the outlet. In order to avoid

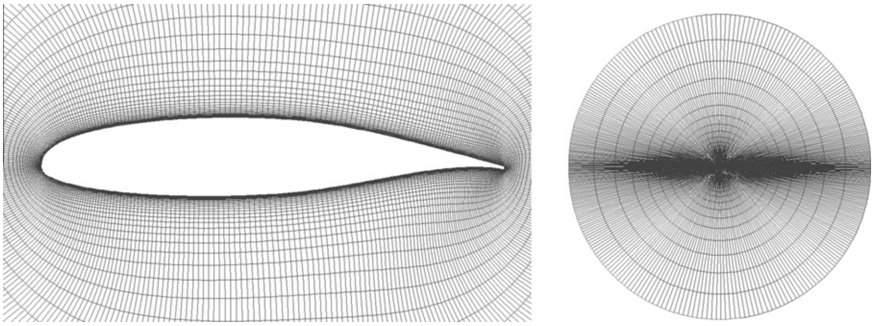

Fig. 3. Computational grid for the GA(W)-1 airfoil.

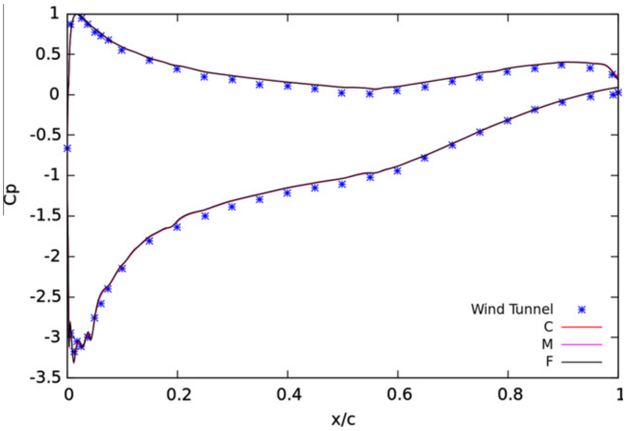

Fig. 4. Experimental (wind tunnel) and numerical pressure coefficient (*Cp*) along the surface of the GA(W)-1 airfoil with $\alpha = 8.02°$ for coarse (*C*), medium (*M*), and fine (*F*) grids.

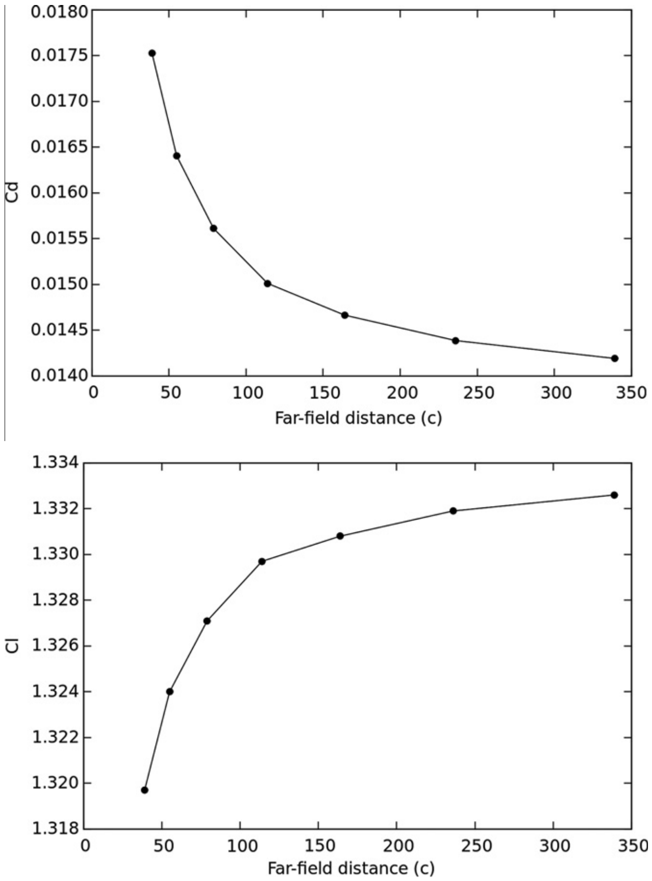

Fig. 5. Drag and lift coefficients (*Cd* and *Cl*, respectively) for the GA(W)-1 airfoil with $\alpha = 8.02°$ as a function of the distance to the boundaries.

inconsistencies in the far field, the points where this division occurs depend on the angle of attack. The no-slip condition is used on the surface of the airfoil.

Results of the grid independence study are presented in Fig. 4, which shows the pressure coefficient (*Cp*) along the profile with $\alpha = 8.02°$. This angle is chosen due to the availability of experimental data and because the maximum lift-to-drag ratio occurs in this region. The three grids give the same results, as shown by the superposition of *C*, *M*, and *F* curves in Fig. 4. Since *Cp* values show good agreement with the experimental data, grid *C* is used for all subsequent simulations, to avoid excessive computational time.

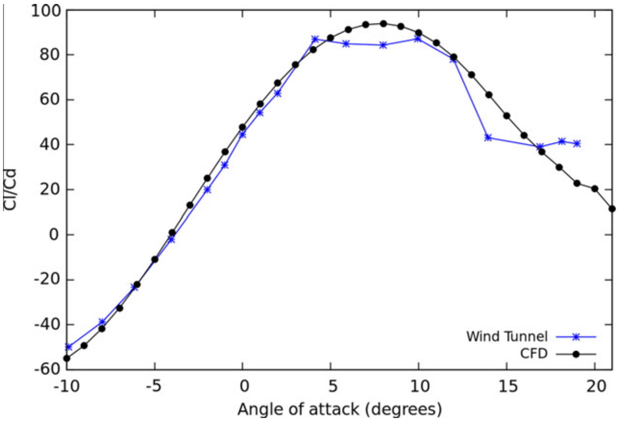

Fig. 6. Experimental (wind tunnel) and numerical (CFD) lift-to-drag ratio (*Cl*/*Cd*) for the GA(W)-1 airfoil.

3.2. Size of the computational domain

A distance of about 100*c* between the obstacle and the boundaries of the fluid domain is recommended by Mavriplis et al. [42]. To verify the possibility of reducing this distance, which reduce computational time, and also to make sure that the distance suggested is enough, simulations are performed with various different domain sizes. The lift and drag coefficients (*Cl* and *Cd*, respectively) as a function of the domain size are presented in Fig. 5.

It can be observed that *Cl* changes very slightly with the increase or decrease of the domain size, but *Cd* requires a larger domain than initially thought to be independent of the far field. With that in mind, the grid to be used in the subsequent simulations has a distance between the airfoil and the external boundaries of about 340*c*.

3.3. Reliability of the aerodynamic coefficients

Since the parameters used in the optimization processes performed in this work are the aerodynamic coefficients *Cl* and *Cd*, it is essential to properly predict these coefficients. The maximum lift-to-drag ratio must also be correct, even for high values of the angle of attack (α). Fig. 6 shows *Cl*/*Cd* as a function of α. Simulations appear to be not only reliable for angles of attack up to that which gives the maximum lift-to-drag ratio, but they also give smoother results than the wind tunnel, which is important for finding maximum values more efficiently.

4. Applications

4.1. Parallelization

Due to the nature of the GA, it can be parallelized in a way that each processor of the computer calculates one individual independently, while the sequential characteristic of gradient-based methods allows only one simulation to be performed at any given time. The GA is parallelized with OpenMP, which is an application programming interface of simple implementation, requiring only small changes in the original code [43]. This is compared with the parallel solver of the commercial software Fluent without modifications. Hence, two options are available: launching multiple solvers at the same time or launching one parallelized solver only. The speed-up for both methods is shown in Fig. 7.

The use of multiple solvers is quite efficient, since each processor works with its own data and only the final results are shared, at the end of the process. The approach of using one parallelized solver only is not as efficient, since the loading time of the software becomes quite long with the increase in the number of processors and that is not compensated by the increase in speed of the solution process. Hence, with 4 processors, the multiple simultaneous solvers approach used in the GA gives a speed-up of about 3.4, while the parallel solver used sequentially gives a speed up of about 1.5, suggesting that if multiple processors are available, GAs can be considerably more efficient than sequential methods.

4.2. Multi-objective optimization with the GA

In order to investigate the impact of the airfoil shape on *Cl*, *Cd*, and *Cl*/*Cd*, a multi-objective optimization with the GA is performed to maximize *Cl* and minimize *Cd*. Eight variables are used to represent the airfoil, as shown in Fig. 2. The angle of attack is the 9th and final variable. To maintain a realistic shape of the airfoil, constraints on its thickness are employed. The thickness in the region corresponding to the maximum coordinate of the extrados is allowed to have a minimum value of

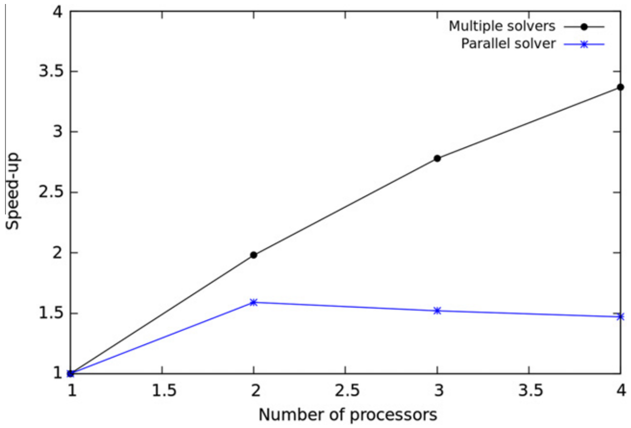

Fig. 7. Computational speed-up with the two parallelization approaches tested.

about 0.17*c*, which is the thickness of the GA-W(1) airfoil. On the position 0.9*c*, a minimum thickness of 0.025*c* is set, to avoid unrealistically thin trailing edges.

The GA requires limits for the input variables, which may be troublesome. After some tests, it was noticed that the variables tended to converge to the extreme values. Thus, the search space was successively increased and displaced, in order to give enough flexibility to the optimization process to find the optimum airfoil. The disadvantage of this process is that a larger search domain means a slower convergence of the optimization procedure. The angle of attack is allowed to vary between 5.5 and 7.5 and the other variables limits are shown in Fig. 8, along with their values for the GA(W)-1.

The algorithm is performed for 50 generations of 28 individuals. The Pareto front and the most relevant airfoils obtained are presented in Fig. 9. It can be seen that the airfoil with maximum *Cl*/*Cd* has a large camber, which is more similar to the airfoil with maximum lift than the airfoil with minimum drag, which tends to a more symmetrical shape. To further improve the lift-to-drag ratio, a single-objective optimization is also carried out.

4.3. Single-objective optimization with the GA and ANN

The objective function to be maximized is *Cl*/*Cd*. The same limits for the input variables and restrictions of the multi-objective optimization are used. The GA is used until the generation *i* with the stop criteria:

$$\begin{aligned}&\frac{std(fobj_i)}{\min(fobj_i)} < 10^{-4},\\&\frac{\min(fobj_i)-\min(fobj_{i-1})}{\min(fobj_i)} < 10^{-4},\end{aligned} \tag{2}$$

where *std* stands for standard deviation and *fobj* is the objective function. The number of generations required to fulfill these conditions is 19. After performing the optimization process using CFD, the data of the first 10 generations are normalized and used to train an artificial neural network, which is then used coupled with the GA as a surrogate model, replacing the CFD simulations. The resulting algorithm is then used for 19 generations to compare the results with the original one.

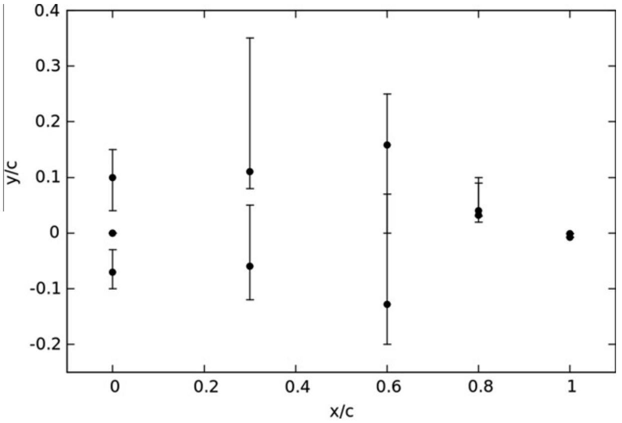

Fig. 8. Variable limits for the genetic algorithm, circles represent the points of the GA(W)-1 and error bars the range of the variables for the optimization.

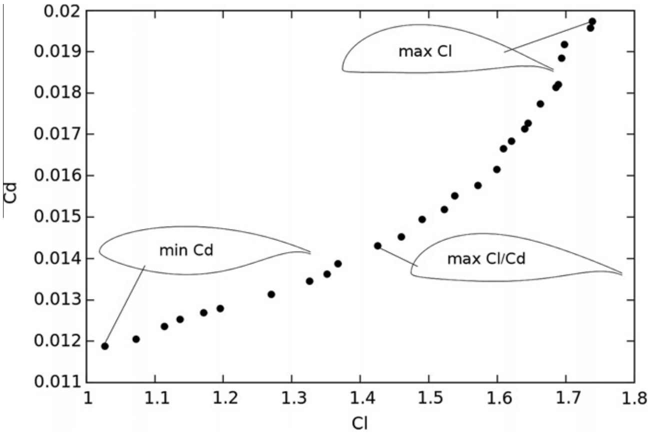

Fig. 9. Pareto front with the fittest airfoils represented.

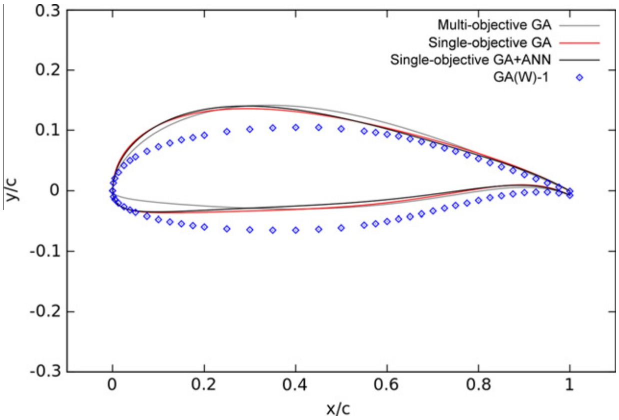

Fig. 10. Airfoils with highest lift-to-drag ratios found with the genetic algorithm (GA) with and without the use of artificial neural networks (ANN) and the GA(W)-1.

Table 1
Highest values of *Cl*/*Cd* found with each optimization procedure.

	Multi-objective GA	Mono-objective GA	Mono-objective GA+ANN
Cl/Cd	100.50	101.13	101.42

The time required to train the ANN and run the coupled GA is smaller than the time necessary to perform the CFD analysis of a single airfoil. Hence, when using the ANN, most of the computational time is spent in generating the training data with CFD, which in the present case involves 10 generations. The total time is therefore practically half of that required using CFD for all 19 generations. The fittest airfoil from the optimization performed with the GA coupled with the ANN was validated with a CFD simulation. Results differ in about 6%. The obtained airfoils are presented in Fig. 10.

The shape of the airfoils obtained with both procedures is very similar, having a larger leading edge radius than the airfoil with highest lift-to-drag ratio found with multi-objective optimization. Values of *Cl*/*Cd* for the three airfoils are reported in Table 1.

5. Conclusions

The following conclusions can be drawn from the present work:

(1) The 2D steady state incompressible Reynolds-averaged Navier–Stokes simulations with the Spalart–Allmaras turbulence model were effective for the case studied here, though the domain size had to be significantly larger than what is indicated in literature.

(2) Parallelization using several separated solvers was shown to be much more efficient than that of a single solver using several processors, probably due to the fact that the grid used in the present work is somewhat small.
(3) Studies on the variable limits for the genetic algorithm are fundamental to avoid convergence to extreme values or excessively large search domains.
(4) The airfoils obtained with single and multi-objective genetic algorithm were similar in shape and aerodynamic coefficients, with a lift-to-drag ratio about 100. The airfoils with maximum lift-to-drag ratio were very similar to the one with maximum lift, which in turn were quite different from the one with minimum drag.
(5) Artificial neural networks were able to model airfoils with large angles of attack, decreasing the computational time by almost 50%.

The technique described here can easily be repeated for different constraints and Reynolds numbers, in order to design airfoils for different sections of a wind turbine blade. Using a computer with 4 processors of 2.8 GHz and 4 GB of RAM, the computational time required, when the surrogate model was employed, was less than 10 h. This technique is a first step towards the optimization of full wind turbines, which is currently being developed.

Acknowledgements

The authors wish to thank the National Council of Scientific and Technological Development (CNPq) for the financial support.

References

[1] K.C. Giannakoglou, Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence, Prog. Aerosp. Sci. 38 (2002) 43–76.
[2] M. Nemec, D.W. Zingg, T.H. Pulliam, Multipoint and Multi-Objective Aerodynamic Shape Optimization, AIAA J. 39 (6) (2004) 1057–1065.
[3] S. Peigin, B. Epstein, Robust optimization of 2D airfoils driven by full Navier–Stokes computations, Comput. Fluids 33 (2004) 1175–1200.
[4] A. Shahrokhi, A. Jahangirian, Airfoil shape parameterization for optimum Navier–Stokes design with genetic algorithm, Aerosp. Sci. Technol. 11 (2007) 443–450.
[5] K. Chiba, S. Obayashi, K. Nakahashi, H. Morino, High-Fidelity Multidisciplinary Design Optimization of Aerostructural Wing Shape for Regional Jet, AIAA Paper, 2005-5080.
[6] N. Kroll, N.R. Gauger, J. Brezillon, R. Dwight, A. Fazzolari, D. Vollmer, K. Becker, H. Barnewitz, V. Schulz, S. Hazra, Flow simulation and shape optimization for aircraft design, J. Comput. Appl. Math. 203 (2007) 397–411.
[7] P. Fuglsang, H.A. Madsen, Optimization method for wind turbine rotors, J. Wind Eng. Ind. Aerodin. 80 (1999) 191–206.
[8] P. Devinant, T. Laverne, J. Hureau, Experimental study of wind-turbine airfoil aerodynamics in high turbulence, J. Wind Eng. Ind. Aerodin. 90 (2002) 689–707.
[9] T. Burton, D. Sharpe, N. Jenjins, E. Bossanyi, Wind Energy Handbook, John Wiley & Sons, West Sussex, 2001.
[10] R.J. McGhee, W.D. Beasley, Low-speed aerodynamic characteristics of 17-percent-thick airfoil section designed for general aviation applications, NASA TN D-7428 (1973).
[11] M.O.L. Hansen, Aerodynamics of Wind Turbines. 2nd ed. Earthscan, London, 2008.
[12] M. Jureczko, M. Pawlak, A. Mezyk, Optimisation of wind turbine blades, J. Mater. Process. Technol. 167 (2–3) (2005) 463–471.
[13] X. Wang, W.Z. Shen, J.Z. Wei, N.S. Jens, J. Chen, Shape optimization of wind turbine blades, Wind Energy 12 (8) (2009) 781–803.
[14] S. Simms, S. Schreck, M. Hand, L.J. Fingersh, NREL Unsteady aerodynamics experiment in the nasa-ames wind tunnel: a comparison of predictions to measurements, NREL/TP-500-29494, June, 2001.
[15] H. Safikhani, A. Khalkhali, M. Farajpoor, Pareto based multi-objective optimization of centrifugal pumps using CFD neural networks and genetic algorithms, Eng. Appl. Comput. Fluid Mech. 5 (1) (2011) 37–48.
[16] M.O.L. Hansen, J.N. Sørensen, S. Voutsinas, N. Sørensen, H.Aa. Madsen, State of the art in wind turbine aerodynamics and aeroelasticity, Prog. Aerosp. Sci. 42 (2006) 285–330.
[17] L. Huyse, S.L. Padula, R.M. Lewis, W. Li, Probabilistic approach to free-form airfoil shape optimization under uncertainty, AIAA J. 40 (9) (2002) 1764–1772.
[18] F. Zhang, S. Chen, M. Khalid, Multi-Point Optimization of Transonic Wing by Real-Coded Genetic Algorithm. In: The Eleventh annual conference of the CFD Society of Canada, Vancouver, May, 2003.
[19] R. Duvigneau, M. Visonneau, Hybrid genetic algorithms and artificial neural networks for complex design optimization in CFD, Int. J. Numer. Methods Fluids 44 (2004) 1257–1278.
[20] M.K. Karakasis, A.P. Giotis, K.C. Giannakoglou, Inexact information aided, low-cost, distributed genetic algorithms for aerodynamic shape optimization, Int. J. Numer. Methods Fluids 43 (2003) 1149–1166.
[21] P. Cinnella, P.M. Congedo, Optimal airfoil shapes for viscous transonic flows of Bethe–Zel'dovich–Thompson fluids, Comput. Fluids 37 (2008) 250–264.
[22] D. López, C. Angulo, L. Macareno, An improved meshing method for shape optimization of aerodynamic profiles using genetic algorithms, Int. J. Numer. Methods Fluids 56 (2008) 1383–1389.
[23] I.C. Kampolis, K.C. Giannakoglou, A multilevel approach to single-and multiobjective aerodynamic optimization, Comput. Methods Appl. Mech. Eng. 197 (2008) 2963–2975.
[24] E. Vatandas, I. Özkol, Coupling dynamic mesh technique and heuristic algorithms in 3-D-tapered wing design, Int. J. Numer. Methods Eng. 74 (2008) 1771–1794.
[25] H.-J. Kim, D. Sasaki, S. Obayashi, K. Nakahashi, Aerodynamic Optimization of Supersonic Transport Wing Using Unstructured Adjoint Method, AIAA J. 39 (6) (2001) 1011–1020.
[26] B. Yin, D. Xu, Y. An, Y. Chen, Aerodynamic optimization of 3D wing based on iSIGHT, Appl. Math. Mech. 29 (5) (2008) 603–610 (Engl. Ed.).
[27] E.J. Whitney, M. Sefrioui, K. Srinivas, J. Periaux, Advances in hierarchical parallel evolutionary algorithms for aerodynamic shape optimization, JSME Int. J. B 45 (1) (2002) 23–28.
[28] M.M. Rai, N.K. Madavan, Aerodynamic Design Using Neural Networks, AIAA J. 38 (1) (2000) 173–182.
[29] M.J. Garcia, P. Boulanger, S. Giraldo, CFD Based Wing Shape Optimization Through Gradient-Based Method. In: Proceedings of the International Conference on Engineering Optimization, Rio de Janeiro, June, 2008.
[30] S. Obayashi, Aerodynamic optimization with evolutionary algorithms. In: Proceedings of the IEEE International Conference on Control, Dearborn, September, 1996.

[31] H.-S. Chung, S. Choi, J.J. Alonso, Supersonic business jet design using a knowledge-based genetic algorithm with an adaptive, unstructured grid methodology, AIAA Paper 2003-3791.
[32] N.F. Foster, G.S. Dulikravich, Three dimensional aerodynamic shape optimization using genetic and gradient search algorithms, J. Spacecr. Rocket. 34 (1) (1997) 36–42.
[33] A. Vicini, D. Quagliarella, Airfoil and wing design through hybrid optimization strategies, AIAA Paper 98-2729.
[34] J.M. Matias, A. Vaamonde, J. Taboada, W. Gonzalez, Comparison of Kriging and neural networks with application to the exploitation of a slate mine, Math. Geol. 36 (4) (2004) 436–486.
[35] P.R. Spalart, S.R. Allmaras, A one-equation turbulence model for aerodynamic flows, AIAA Paper 92-0439.
[36] S.V. Patankar, Numerical Heat Transfer and Fluid Flow, Hemisphere Publishing Corporation, Washington, 1980.
[37] B.P. Leonard, A stable and accurate convective modelling procedure based on quadratic upstream interpolation, Comput. Methods Appl. Mech. and Eng. 19 (1) (1979) 59–98.
[38] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison Wesley, Alabama, 1989.
[39] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multi-objective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 181–197.
[40] S. Haykin, Neural Networks: A Comprehensive Foundation, second ed., Prentice-Hall, Upper Saddle River, 1999.
[41] S. Nissen, Implementation of a Fast Artificial Neural Network Library (FANN), Technical Report, Department of Computer Science University of Copenhagen (DIKU), 2003.
[42] D. Mavriplis, J.C. Vassberg, E.N. Tinoco, M. Mani, O.P. Brodersen, B. Eisfeld, R.A. Wahls, J.H. Morrison, T. Zickuhr, D. Levy, M. Murayama, Grid Quality and Resolution Issues from the Drag Prediction Workshop Series, J. Aircr. 46 (3) (2009) 935–950.
[43] B. Chapman, G. Jost, R. Van Der Pas, Using OpenMP: Portable Shared Memory Parallel Programming, tenth ed., MIT Press, Cambridge, 2008.