AN AIRFOIL OPTIMIZATION TECHNIQUE FOR WIND TURBINES

André F. P. Ribeiro, ribribeiro@yahoo.com.br
Armando M. Awruch, amawruch@ufrgs.br
Herbert M. Gomes, herbert@mecanica.ufrgs.br
Graduate Program in Mechanical Engineering, Federal University of Rio Grande do Sul, Porto Alegre, RS, Brazil

Abstract. *Optimization algorithms coupled with computational fluid dynamics are used for an airfoil to be employed in wind turbines. 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. Simulations are performed with the incompressible Reynolds-averaged Navier-Stokes equations in steady state using a one equation turbulence model, which are validated with experimental results. Different approaches for parallelization of the computational code are addressed. Mono and multi-objective genetic algorithms are employed. Artificial neural networks are used as a surrogate model and decrease the computational time in about 50%.*

Keywords*: computational fluid dynamics, genetic algorithms, artificial neural networks*

1. INTRODUCTION

With the growth of the importance of renewable energy, airfoil design specifically for wind turbines has become an important issue. Although airfoil optimization with genetic algorithms has been performed by several different groups in the past decade (*e.g*. Giannakoglou, 2002, Nemec *et al*., 2004, Peigin and Epstein, 2004, Shahrokhi and Jahangirian, 2007), they are usually applied having aircrafts in mind. Optimization of entire aircrafts have been made using computational fluid dynamics (CFD) and different types of algorithms (*e.g*. Chiba *et al*., 2005, Kroll *et al.*, 2007), while wind turbines are commonly only optimized using the blade element theory, without changing the airfoils used or the general shape of the blade (Fuglsang and Madsen, 1999).

Frequently, airfoils created for aircrafts are used in wind turbines (Devinant *et al*., 2002), even though the design criteria are not the same for both cases. For instance, in wind turbines the airfoil can be used at a much higher angle of attack and the most important parameter is the lift-to-drag ratio (Burton *et al*., 2001). In aircrafts, the objective of the design process might be quite different, for instance, to decrease the drag for a fixed lift coefficient (McGhee and Beasley, 1973). Another concern is that Reynolds and Mach numbers are much lower for wind turbines than they are for aircrafts, which may change the flow significantly. 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.

2. METHODS USED IN THE AERODYNAMIC OPTIMIZATION

A cycle of the optimization procedure, displaying the involved steps, is shown in Fig. 1. 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 airfoils, which are then utilized in numerical simulations in order to obtain the objective function, that must be given back to the optimization algorithm. 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, to perform the optimization, and the employed surrogate model are presented in the following sections.

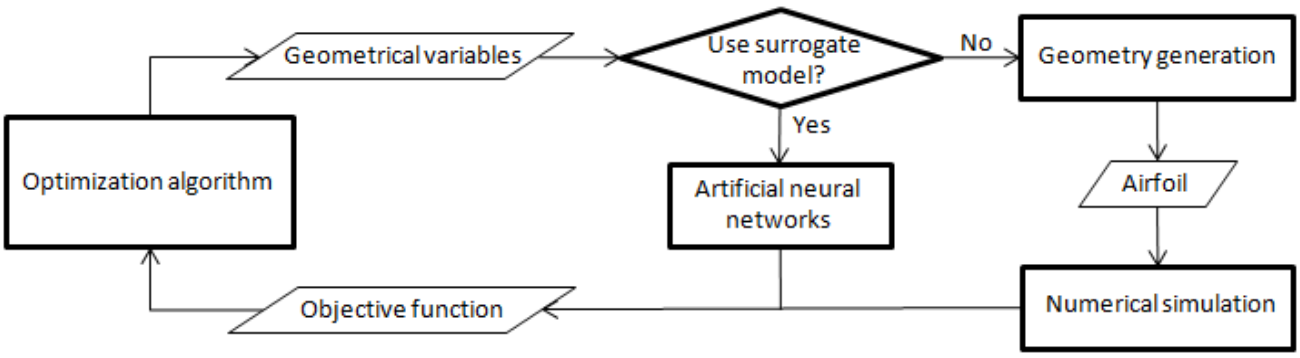

Figure 1. Flow chart of the optimization cycle employed

2.1. Geometry generation

An important part of the optimization process is the parameterization of the geometry, since it is the way one translates the variables generated by the algorithm into the actual airfoil. According to Giannakoglou (2002), 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) done so that its design variables are directly linked to the constraints.

Bézier were chosen for this work, because they are widely accepted in aerodynamic optimization (Karakasis et al., 2003, Peigin and Epstein, 2004, Cinnella and Congedo, 2008, López et al., 2008, and Vatandas and Özkol, 2008) and satisfy the above points, 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.

Figure 2 shows a GA(W)-1 airfoil (McGhee and Beasley, 1973) with chord c generated using two Bézier curves, one for the extrados and one for the intrados, each with 6 points, and the points used to generate them.

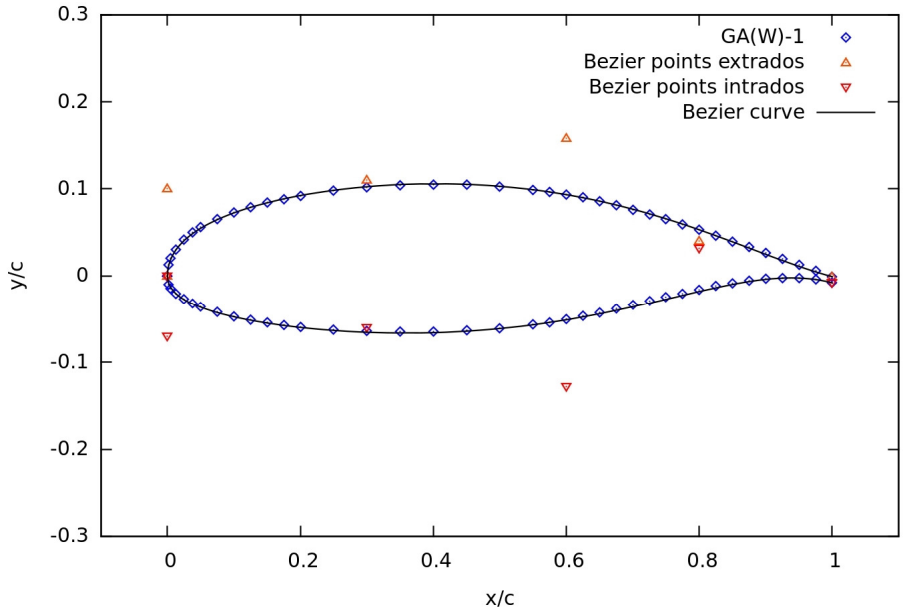

Figure 2. GA(W)-1 airfoil approximated with Bézier curves

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 (Spalart and Allmaras, 1992), as it is often employed in aerodynamic optimization of airfoils and aircrafts (Chiba *et al*., 2005, Kampolis and Giannakoglou, 2008, and Yin *et al*., 2009). Numerical simulations are performed with the finite volume, cell-centered commercial code Fluent. The pressure-velocity coupling is done with the Semi Implicit Linked Equations (SIMPLE, Patankar, 1980) algorithm. Central differences are employed to interpolate the pressure on cell faces and Quadratic Upwind Interpolation for Convective Kinematics (QUICK, Leonard, 1979) is used to interpolate the convective terms of the other variables.

2.3. Genetic algorithms

According to Vicini and Quagliarella (1998), 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 (GA, Goldberg, 1989) are based on Darwin's evolution theory, as they take an initial, randomly generated population and allow the fittest individuals to reproduce, with crossover, mutation, and elitism involved in this process. They are used in 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, specially gradient-based methods.

In the present study, the open source software NSGA-II (Deb et al., 2002) 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 (Haykin, 1999). ANNs have been successfully employed to represent airfoils at low angles of attack, *e.g.* Rai and Madavan (2000) and Cinnella and Congedo (2008).

The ANN library used in this work is FANN (Fast Artificial Neural Networks). It is an open source project started by Nissen (2003). 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 used.

3. VALIDATION OF THE NUMERICAL SIMULATIONS

To guarantee the reliability of the numerical simulations performed in this work, a validation process was carried out. The airfoil used for the validation was the GA(W)-1 (McGhee and Beasley, 1973), due to the fact that it has detailed experimental results for relatively low speeds and high angles of attack (α). The numerical results were 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 started with the generation of three 2D grids: a coarse (*C*), a medium (*M*), and a fine (*F*) mesh. These grids were created following the recommendations of Mavriplis *et al.* (2009), with the following characteristics:

1) in the boundary layer region, $y+ \leq [1,2/3,4/9][C,M,F]$, 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 geometry;
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 between the different grids.

Mavriplis *et al.* (2009) also suggest a fourth, extra fine (*XF*) grid, but this was not shown to be necessary in the present case. The high quality structured O-grid used is shown in Fig. 3. The far field was divided into two regions so that the inlet could have a prescribed velocity boundary condition, while at the outlet pressure is prescribed. The points where this division occurs depend on the angle of attack to avoid inconsistencies in the far field. The no-slip condition was used on the airfoil itself.

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

3.2. Size of the computational domain

Mavriplis *et al.* (2009) recommend a distance between the geometry and the boundaries of about $100c$. To verify the possibility of reducing that distance, which would give a reduction in computation time, and also to make sure that the distance suggested is enough, simulations were 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.

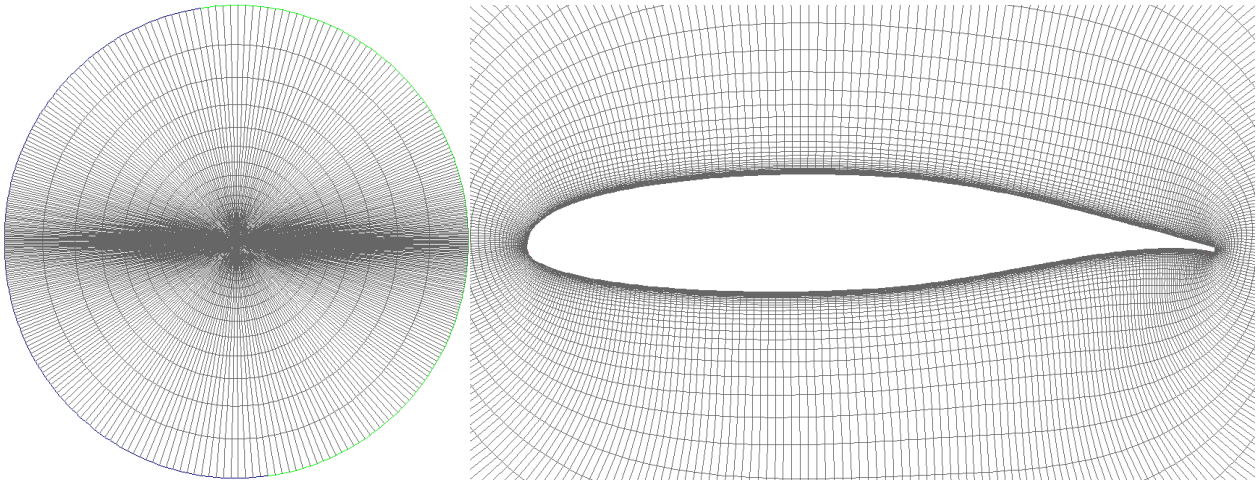

Figure 3. Computational grid for the GA(W)-1 airfoil

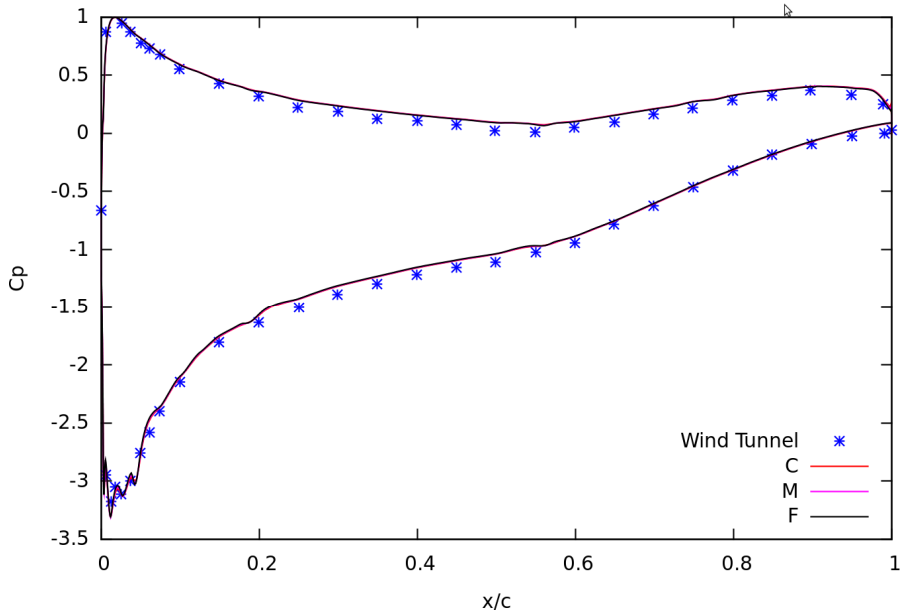

Figure 4. Experimental (Wind Tunnel) and numerical pressure coefficient (*Cp*) along the surface of the GA(W)-1 airfoil with α =8.02° for Coarse (*C*), Medium (*M*), and Fine (*F*) grids

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 chosen to be used in the optimizations has a distance between the airfoil and the boundaries of about 340*c*.

3.2. Reliability of the aerodynamic coefficients

Since the parameters used in the optimizations performed in this work are the aerodynamic coefficients *Cl* and *Cd*, it is fundamental to have reliable predicted values for them. The maximum lift-to-drag ratio must also be reliable, even for high values of the angle of attack (α). Figure 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.

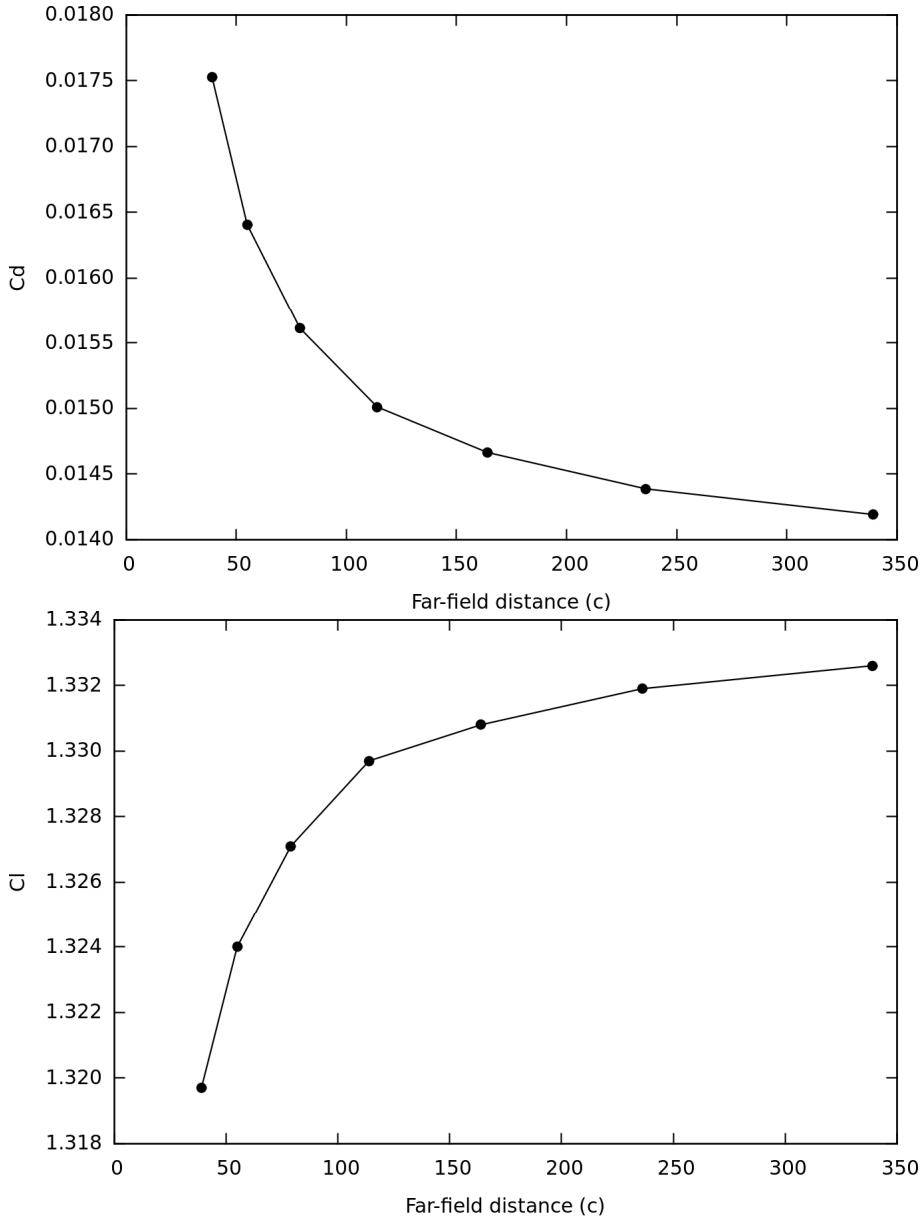

Figure 5. Drag and lift coefficients (*Cd* and *Cl*, respectively) for the GA(W)-1 airfoil with α=8.02° as a function of the distance to the boundaries

4. APPLICATIONS

4.1. Parallelization

Due to the nature of the GA, it can be parallelized in a way where each processor of the computer calculates one individual independently, while the sequential characteristic of gradient-based methods allow only one simulation to be performed at any given time. The GA was parallelized with OpenMP, which is an application programming interface of simple implementation, requiring only small changes in the original code (Chapman *et al.*, 2008). This was compared with Fluent's parallel solver without modifications. Hence, two options were available: launching multiple solvers at the same time or launching only one parallelized solver. 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 final results are shared, at the end of the process. The approach of using only one parallelized solver was not as efficient, since the loading time of the software became quite long with the increase in the number of processors. Hence, with 4 processors, the multiple simultaneous solvers approach used in the GA gave a speed-up of about 3.4, while the parallel solver used sequentially gave a speed up of about 1.5, which indicates that if multiple processors are available, GAs may be much more efficient than sequential methods.

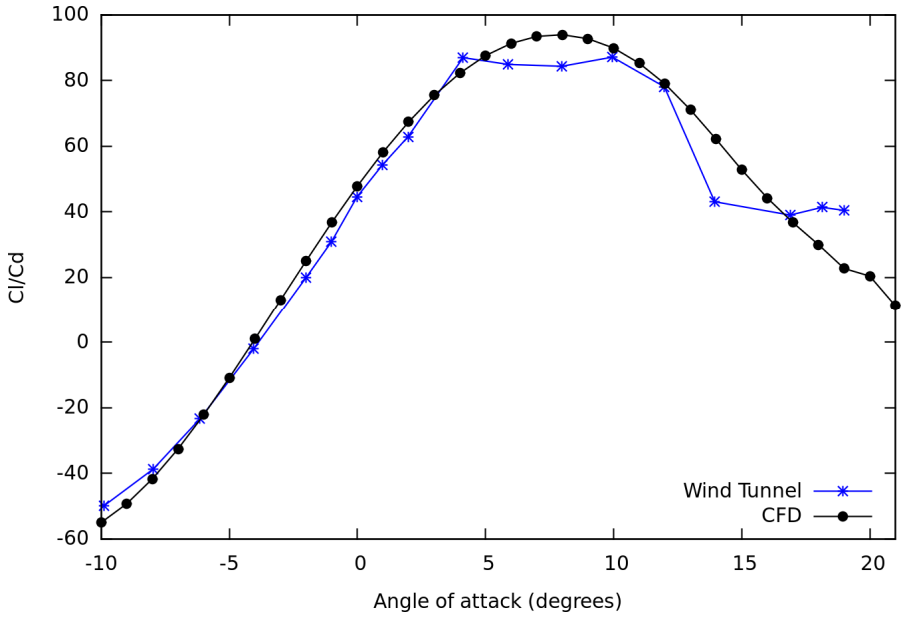

Figure 6. Experimental (Wind Tunnel) and numerical (CFD) lift-to-drag ratio (*Cl*/*Cd*) for the GA(W)-1 airfoil

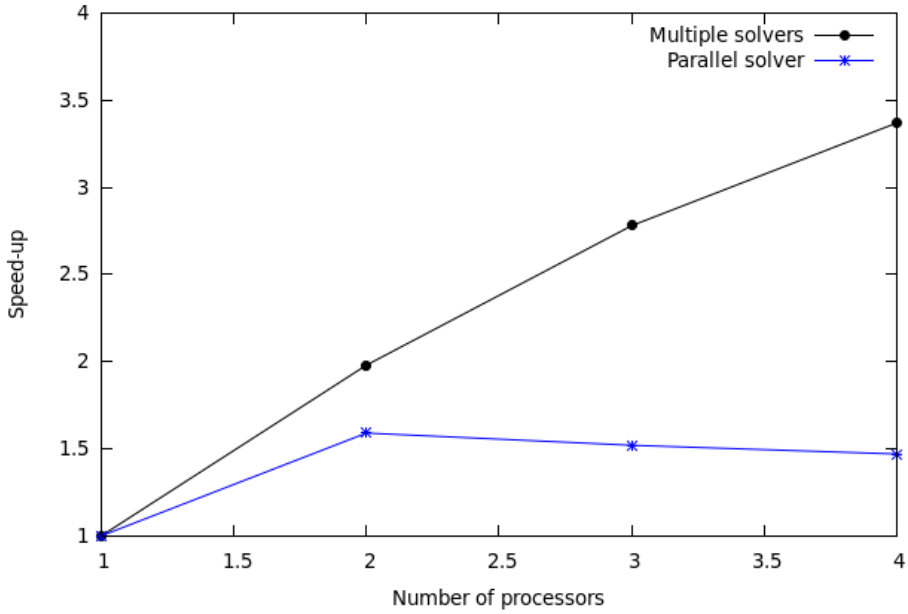

Figure 7. Computational speed-up with the two approaches to parallelization tested

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 was performed to maximize *Cl* and minimize *Cd*. Eight variables were used to represent the airfoil, as in Fig. 2. The angle of attack was the 9th and last variable. To keep a realistic shape of the airfoil, constraints on the thickness were employed. The thickness in the region corresponding to the maximum coordinate of the extrados was allowed to have a minimum value of 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* was set, to avoid unrealistically thin trailing edges.

The GA requires limits for the input variables, which may be troublesome. After few 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 the optimization enough flexibility 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 was allowed to vary between 5.5 and 7.5 and the other variables limits are shown in Figure 8, along with their values for the GA(W)-1.

The algorithm was 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 mono-objective optimization was also carried out.

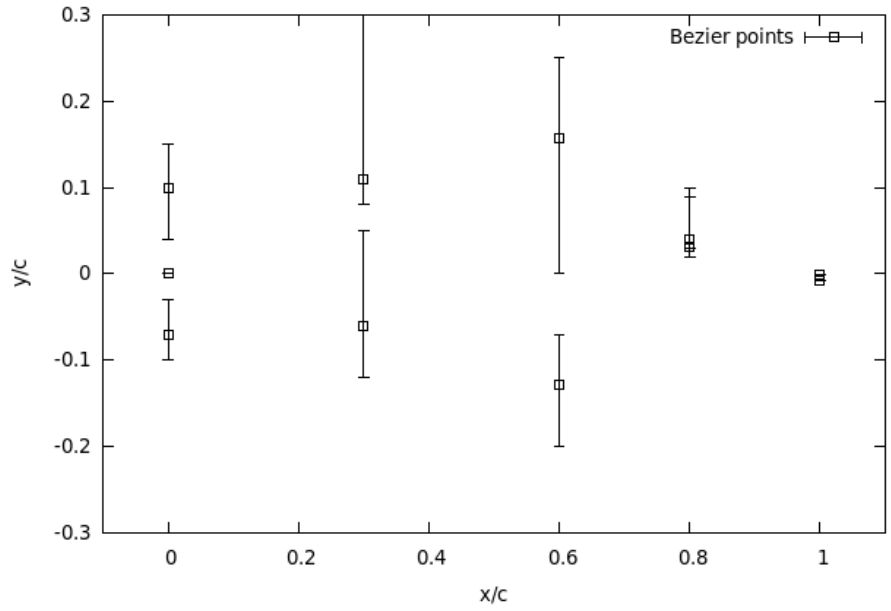

Figure 8. Variable limits for the genetic algorithm, squares represent the points of the GA(W)-1 and error bars the range of the variables for the optimization

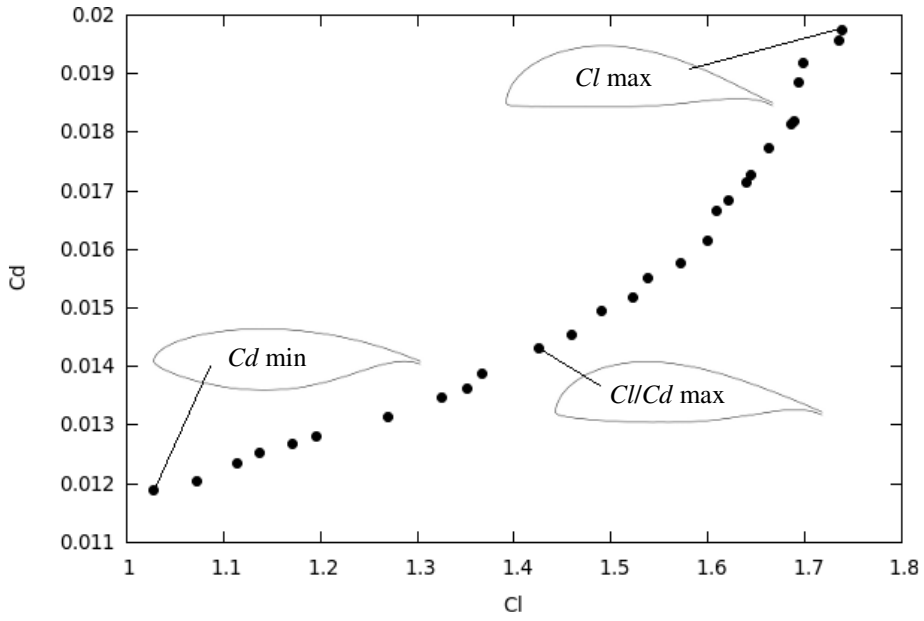

Figure 9. Pareto front with the fittest airfoils represented

4.3. Mono-objective optimization with the GA and ANN

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

$$\frac{std(fobj_i)}{\min(fobj_i)} < 10^{-4}$$
$$\frac{\min(fobj_i) - \min(fobj_{i-1})}{\min(fobj_i)} < 10^{-4} \quad (2)$$

where *std* stands for standard deviation and *fobj* is the objective function. The number of generations required to fulfill these conditions was 19.

After optimizing using only the GA, the data of the first 10 generations were normalized and used to train an artificial neural network, which was then used coupled with the GA as a surrogate model replacing the CFD simulations. The resulting algorithm was used for 19 generations to compare the results with the original one. The training was done using a back propagation algorithm.

The time required to train the ANN and run the GA coupled with it was less than the time necessary to perform the CFD analysis of a single airfoil. Hence, when using the ANN, the computational time is mostly spent on the generation of the training data with CFD, which in this case were 10 generations. The total time was therefore practically half of that required using only the GA for 19 generations, since CFD simulations were done for only 10 generations. The fittest airfoil from the optimization performed with the GA coupled with the ANN was validated with a CFD simulation. Results differed in about 6%. The obtained airfoils are presented in Fig. 10.

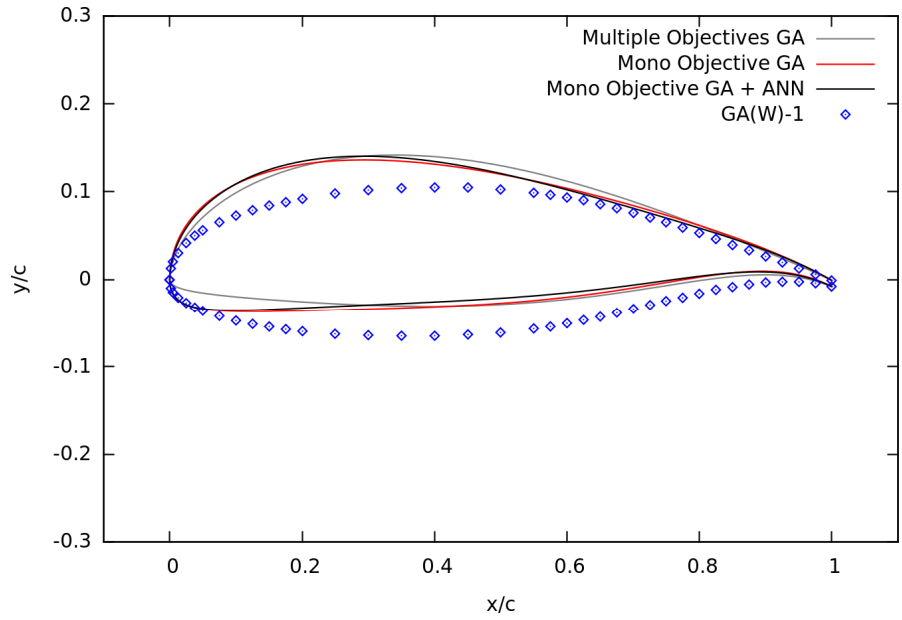

Figure 10. Airfoils with highest lift-to-drag ratios found with the genetic algorithm (GA) with and without the use of artificial neural networks (ANN) and the GA(W)-1

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.

Table1. 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

4. 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 very effective for the tested case;

2) parallelization using several separated solvers was shown to be much more efficient than that of a single solver using several processors;

3) studies on the variable limits for the genetic algorithm are fundamental to avoid convergence to extreme values or an excessively large search domain;

4) the airfoils obtained with mono and multi-objective genetic algorithm were similar in shape and aerodynamic coefficients, with a lift-to-drag ratio about 100;

5) artificial neural networks were able to model airfoils with large angle of attack, decreasing the computational time by about 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 hours.

5. ACKNOWLEDGEMENTS

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

6. REFERENCES

Burton, T., Sharpe, D., Jenjins, N., Bossanyi, E., 2001, "Wind Energy Handbook", Ed. John Wiley & Sons, West Sussex, 617 p.

Chapman, B., Jost, G., Van Der Pas, R., 2008, "Using OpenMP: portable shared memory parallel programming", 10th Ed. Cambridge, MIT Press, 21 p.

Chiba, K., Obayashi, S., Nakahashi, K., Morino, H., 2005, "High-Fidelity Multidisciplinary Design Optimization of Aerostructural Wing Shape for Regional Jet", AIAA Paper 2005-5080.

Cinnella, P., Congedo, P.M., 2008, "Optimal airfoil shapes for viscous transonic flows of Bethe-Zel'dovich-Thompson fluids", Computers and Fluids, Vol. 37, pp. 250-264.

Deb, K., Pratap. A., Agarwal, S., Meyarivan, T., 2002, "A fast and elitist multi-objective genetic algorithm: NSGA-II", IEEE Transaction on Evolutionary Computation, Vol. 6, No. 2, pp. 181-197.

Devinant, P., Laverne, T., Hureau, J., 2002, "Experimental study of wind-turbine airfoil aerodynamics in high turbulence", Journal of Wind Engineering and Industrial Aerodynamics, Vol. 90, pp. 689-707.

Fuglsang, P., Madsen, H.A., 1999, "Optimization method for wind turbine rotors", Journal of Wind Engineering and Industrial Aerodynamics, Vol. 80, pp. 191-206.

Giannakoglou, K.C., 2002, "Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence", Progress in Aerospace Sciences, Vol. 38, pp. 43-76.

Goldberg, D.E., 1989, "Genetic algorithms in search, optimization and machine learning", Ed. Addison Wesley, Alabama, 413 p.

Haykin, S., 1999, "Neural Networks: a comprehensive foundation", 2nd Ed. Prentice-Hall, Upper Saddle River, 842 p.

Kampolis, I.C., Giannakoglou, K.C., 2008, "A multilevel approach to single-and multiobjective aerodynamic optimization", Computer methods in applied mechanics and engineering, Vol. 197, pp. 2963-2975.

Karakasis, M.K., Giotis, A.P., Giannakoglou, K.C., 2003, "Inexact information aided, low-cost, distributed genetic algorithms for aerodynamic shape optimization", International Journal For Numerical Methods in Fluids, Vol. 43, pp. 1149-1166.

Kroll, N., Gauger, N.R., Brezillon, J., Dwight, R., Fazzolari, A., Vollmer, D., Becker, K., Barnewitz, H., Schulz, V., Hazra, S., 2007, "Flow simulation and shape optimization for aircraft design", Journal of Computational and Applied Mathematics, Vol. 203, pp. 397-411.

Leonard, B.P., 1979, "A stable and accurate convective modelling procedure based on quadratic upstream interpolation", Computer Methods in Applied Mechanics and Engineering, Vol. 19, No. 1, pp. 59-98.

López, D., Angulo, C., Macareno, L., 2008, "An improved meshing method for shape optimization of aerodynamic profiles using genetic algorithms", International Journal For Numerical Methods in Fluids, Vol. 56, pp. 1383-1389.

Mavriplis, D., Vassberg, J.C., Tinoco, E.N., Mani, M., Brodersen, O.P., Eisfeld, B., Wahls, R.A., Morrison, J.H., Zickuhr, T., Levy, D., Murayama, M., 2009, "Grid Quality and Resolution Issues from the Drag Prediction Workshop Series", Journal of Aircraft, Reston, Vol. 46, No. 3, pp. 935-950.

McGhee, R.J., Beasley, W.D., 1973, "Low-Speed Aerodynamic Characteristics of 17-Percent-Thick Airfoil Section Designed for General Aviation Applications", NASA TN D-7428.

Nemec, M., Zingg, D.W., Pulliam, T.H., 2004, "Multipoint and Multi-Objective Aerodynamic Shape Optimization", AIAA Journal, Vol. 39, No. 6, pp. 1057-1065.

Nissen, S., 2003, "Implementation of a Fast Artificial Neural Network Library (FANN)", Technical Report, Department of Computer Science University of Copenhagen (DIKU), 88 p.

Patankar, S.V., 1980, "Numerical Heat Transfer and Fluid Flow", Ed. Hemisphere Publishing Corporation, Washington, 197 p.

Peigin, S., Epstein, B., 2004, "Robust optimization of 2D airfoils driven by full Navier-Stokes computations". Computers and Fluids, Vol. 33, pp. 1175-1200.

Rai, M.M., Madavan, N.K., 2000, "Aerodynamic Design Using Neural Networks", AIAA Journal, Vol. 38, No. 1, pp. 173-182.

Shahrokhi, A., Jahangirian, A., 2007, "Airfoil shape parameterization for optimum Navier-Stokes design with genetic algorithm", Aerospace Science and Technology, Vol. 11, pp. 443-450.

Spalart, P.R., Allmaras, S.R. , 1992, "A one-equation turbulence model for aerodynamic flows", AIAA-Paper 92-0439.

Vatandas, E., Özkol, I., 2008, "Coupling dynamic mesh technique and heuristic algorithms in 3-D-tapered wing design", International Journal For Numerical Methods in Engineering, Vol. 74, pp. 1771-1794.
Yin, B., Xu, D., An, Y., Chen, Y., 2008, "Aerodynamic optimization of 3D wing based on iSIGHT", Applied Mathematics and Mechanics (English Edition), Vol. 29, No. 5, pp. 603-610.

7. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.