

# TRANSONIC AIRFOIL SHAPE OPTIMIZATION USING MULTI-OBJECTIVE GENETIC ALGORITHM

**Alexandre P. Antunes**

EMBRAER, Av. Brigadeiro Faria Lima, 2170 S.J. dos Campos - SP  
[alexandre.antunes@embraer.com.br](mailto:alexandre.antunes@embraer.com.br)

**João Luiz F. Azevedo**

CTA/IAE/ASE-N, 12228-904, São José dos Campos, SP  
[azevedo@iae.cta.br](mailto:azevedo@iae.cta.br)

**Luís Carlos de Castro Santos**

EMBRAER, Av. Brigadeiro Faria Lima, 2170 S.J. dos Campos - SP  
[luis.castro@embraer.com.br](mailto:luis.castro@embraer.com.br)

**Abstract.** Genetic Algorithms (GA) are search optimization methods that use principles of natural genetics and natural selection. In this method, the possible solutions for a certain problem in question is represented by some form of a biological population, which evolves over generations to adapt to an environment by selection, crossover and mutation. Instead of working with a single solution at each iteration, GA work with a number of solutions, known as a population. The main goal in a single-objective optimization is simply to find the global optimal solution. On the other hand, in a multi-objective optimization, there is more than one objective function, and each one may have a different optimal solution. If there is sufficient difference in the optimal solutions, the objective functions are conflicting with each other. The multiple optimal solutions exist because no solution can be optimal for multiple conflicting objectives, and the obtained set of possible solutions is known as Pareto-front. Once multiple solutions are found, a higher-level decision is adopted; here one must choose which solution better satisfies the optimization problem in question. Aerodynamic design optimizations are often multimodal and nonlinear problems, because the flowfield is governed by a system of nonlinear partial differential equations. This work is focused on airfoil shape optimization using a multi-objective genetic algorithm concept known as Non-dominated Sorting Genetic Algorithm – NSGA. The MSES code, an Euler equation solver with boundary layer correction, is coupled to the NSGA implementation. The airfoil shape is obtained by polynomial parameterization and, for each of the multiple design points, the drag coefficient ( $C_d$ ) is used as objective function.

**Key Words:** Genetic Algorithms, Multi-Objective, Airfoil Shape Optimization, CFD.

## 1. INTRODUCTION

Since the factitious September 11th, the demand for commercial airplane has diminished all over the world. This fact leads to a harder competition between the aeronautical companies, in such a way that only the fittest can satisfactory respond to the market necessities. In general terms, the best choice means not only a lower price for an airplane acquisition but also a lower direct operating cost (DOC). The crew costs, the maintenance costs (airframe and engine), the insurance costs, and the depreciation of the airplane contribute to the direct operating cost (DOC).

In fact, to obtain a lower DOC, it is necessary to perform an optimization of the whole product, in this case the airplane. In such a process, all the involved design areas participate of this *Multidisciplinary Design Optimization* (MDO) study. Such approach is a present trend in the aeronautical industry and, in the context of the aerodynamic design team, the main goal consists in obtaining a lower drag value at a Mach number as higher as possible. Typically, an increase in the flight Mach number produces a reduction in the DOC, due to the increase in productivity caused by the reduction of the flight time. However, this increase in the Mach number must be accomplished with a satisfactory operation performance; otherwise the airplane might not be competitive anymore.

Although airfoils are two-dimensional representations of wing profiles, their influence on the 3D designs can be very effective. Therefore, the proposed design methodology will be based solely on the profile modifications. It is not always true that these optimized profiles produce an improvement in the wing design, due to the three-dimensional characteristics of the flow. Nevertheless, when a better wing is obtained, we have a considerable reduction in drag since the wing answers for 2/3 of the airplane drag at a typical cruise condition (Nixon, 1981).

This work is an effort to optimize airfoil shapes using the NSGA concept. The optimization is performed at the transonic flow regime and for more than one point of operation. Here a point of operation is defined by a desired value of lift coefficient ( $C_l$ ) and flight Mach number (Mach), at a fixed Reynolds number. The need for a multi-point optimization is based on the fact that, when single point optimized airfoils are evaluated in an off design condition, they frequently have a poor aerodynamic behavior (Antunes *et al.*, 2003). A multi-objective genetic algorithm, namely the non-dominated sorting genetic algorithm (NSGA), was implemented in order to have a reasonable behavior over all the Mach number range of interest by means of a Pareto front concept.

The airfoil shape was obtained by a polynomial parameterization of the thickness and the airfoil camber line, (Streshinsky, 1994). In this parameterization, the polynomial coefficients that describe the thickness and the airfoil camber line are obtained by solving simple and simultaneous equations that satisfy the imposed values for the: leading-

edge radius, upper and lower crest location, curvatures, trailing-edge ordinate, thickness, direction and wedge angle, among others. The genetic algorithm, NSGA, is used to search the best values of these aerodynamic parameters at the desired design points or, in other words, the set of values that minimize or maximize a certain objective function. In the present work two objective functions, based on the drag coefficient (Cd), were evaluated at two different design points.

## 2. THEORETICAL FORMULATION

The theoretical formulation of a multi-objective genetic algorithm, in this case the NSGA, only differs from a single-objective approach by the form that the selection operation is performed. The theoretical formulation of the single-objective procedure, adopted in the present paper, is described in detail in a previous work by the present authors (Antunes *et al.*, 2003). Hence, such formulation will not be repeated here and the interested reader is referred to Antunes *et al.* (2003) for the single-objective algorithm upon which the present multi-objective optimization procedure is built. Prior to the reproduction stage of a NSGA, two steps are added. The first one consists in ranking the population based on the individual's non-domination level; and the second one is the sharing function method, which is used to assign fitness to each individual (Deb, 1999). The non-domination ranking methodology and the sharing function will be described in the forthcoming subsection. Basically, the NSGA works using the following sequence of concepts:

- An initial population of candidates is generated;
- The objective function of the population is evaluated;
- Those elements of the population which satisfy certain criteria are chosen to reproduce to the next generation;
- New 'individuals' are created by the exchange of features from the 'individuals' previously selected;
- Some of the 'individuals' created can suffer mutation.

This process can be represented by the flowchart below.

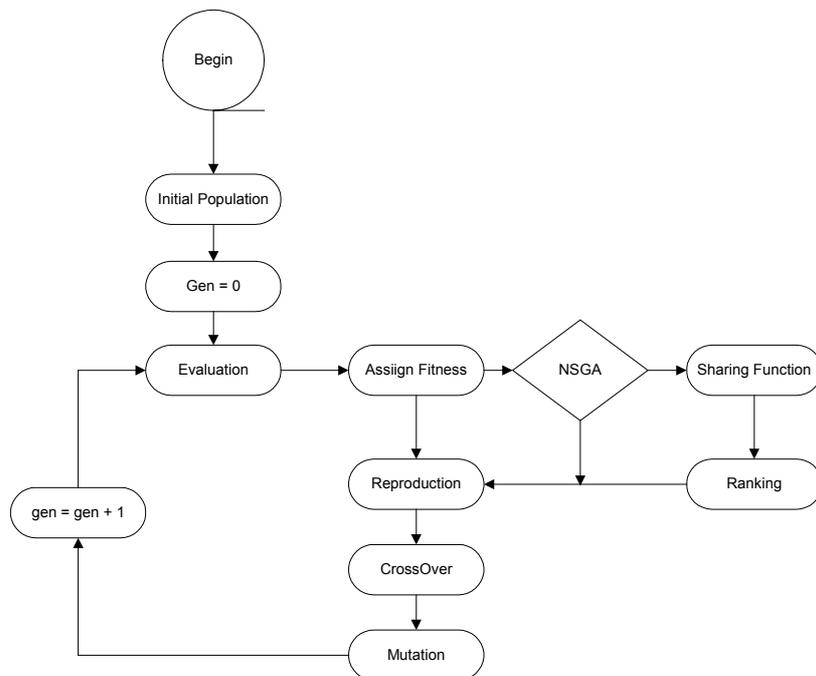


Figure 1. Flowchart of the NSGA concept.

### 2.1 INITIALIZATION

The first step in the GA concept is to decide the strategy of coding and decoding the design variables in binary strings. The design variables in question are the aerodynamic parameters: leading-edge radius, upper and lower crest location, curvatures, trailing-edge ordinate, thickness, direction and wedge angle, among others. The following string is an example of a binary representation of these aerodynamic parameters. In the binary string below we have six coded coefficients,

$$\underbrace{1010}_{a_1} \underbrace{0011}_{a_2} \underbrace{1100}_{a_3} \underbrace{0010}_{a_4} \underbrace{0001}_{a_5} \underbrace{1110}_{a_6} \quad (1)$$

Each aerodynamic coefficient  $a_i$  is represented by a binary string with a certain length. The sum of all these binary parts generates the individual and the set of individual forms the population. The name individual is an analogy with the natural biology and it is responsible for the core of the GA, since they contain the vital information from the design variables.

In the initialization stage, a predecessor population is created. This process consists in the generation of ‘individuals’, in which the binary strings are randomly sorted. The lower binary string is (0000) and the upper binary string is the (1111). These two strings represent the  $x_i^{\min}$  and the  $x_i^{\max}$ , respectively. The value of  $x_i^{\min}$  and  $x_i^{\max}$  are defined as the bounds of the solution domain. Any other sorted binary string is contained within these two exposed binary extremes and the  $x_i$  value for it can be obtained using the following formulae to decode the information:

$$x_i = x_i^{\min} + \frac{x_i^{\max} - x_i^{\min}}{2^{l_i} - 1} DV(a_i) \quad (2)$$

where  $l_i$  is the length of the binary representation of the weight coefficient,  $a_i$ , and  $DV(a_i)$  is the conversion of the binary string that represents the  $a_i$  coefficient to decimal. The length,  $l_i$ , can be computed as follows:

$$l_i = \log_2 \left( \frac{x_i^{\max} - x_i^{\min}}{\varepsilon_i} \right). \quad (3)$$

Here  $\varepsilon_i$  denotes the decimal precision of the weight coefficient,  $a_i$ . Notice that the binary string length of each coefficient is a function of the desired decimal precision. So, the length of the binary string grows with the increase of the precision desired for the coefficient. The total length of the binary string is the sum of the lengths of each binary coefficient representation.

One should notice that this procedure of random sort creates variability in the possible aerodynamic parameters. It creates possible solutions all over the solution domain and only those that are the fittest will pass to the next generation. The two ‘individuals’ below are an example of how different a set of aerodynamic parameters can be produced with this binary random sort process:

$$\begin{array}{l} \underbrace{01000}_{-0.0029} \underbrace{11110}_{0.0056} \text{ Individual 1 ,} \\ \underbrace{01001}_{-0.0025} \underbrace{11101}_{0.0052} \text{ Individual 2 .} \end{array} \quad (4)$$

In this specific example,  $x_i^{\min} = -0.006$ ,  $x_i^{\max} = 0.006$ , and  $\varepsilon_i = 0.00375$  for the two coefficients  $a_i$ . However, one might choose different values of  $x_i^{\min}$ ,  $x_i^{\max}$  and  $\varepsilon_i$  for each one of the  $a_i$  coefficients.

## 2.2 EVALUATION

In this step, an objective function is used to classify the solution represented by the ‘individuals’. The definition of the objective function depends on the problem in question. In this work, the objective function is the drag coefficient (Cd), which is evaluated by the MSES code for a given geometry. This evaluation of Cd is performed for more than one design point as the present work is concerned with a multi-objective optimization. MSES (Drela, 1996) is an Euler solver with a coupled boundary layer routine that takes into account the most relevant viscous effects.

## 2.3 SELECTION

It is a stage in which the ‘individuals’ are selected for the reproduction phase. There is no rule about how many ‘individuals’ must be selected. One can choose to select as many individuals as there were in the initial population or, even, one can allow for having more ‘individuals’ than in the initial population. In the latter case, an increase in population is being allowed. The selection procedure is very important because, depending on the selection method adopted, there is the possibility to create super ‘individuals’ for the next generation, causing loss of diversity. On the other hand, a non-elitist method produces ‘individuals’ that turn the optimization method ineffective. The idea behind NSGA is that a ranking selection method emphasizes the current non-dominated ‘individuals’ and the sharing function method maintains diversity of the population.

In the first step of this NSGA implementation, the non-dominated ‘individuals’ are grouped in fronts based on the Cd coefficient for each design point. In order to obtain these fronts, one has to discover the non-dominated ‘individuals’, which can be identified whenever the two following statements are satisfied:

- The individual  $j$  is non-dominated if it is no worse than individual  $i$  in all the objectives,
- The individual  $j$  is non-dominated if it is strictly better than individual  $i$  in at least one objective.

Using the two statements above, one can generate the non-dominated fronts using the following steps:

- For a population of size  $N$ , where each individual has  $M > 1$  objective functions, begin with the individual  $i = 1$ .
- For all ‘individuals’  $j$  from 1 to  $N$  and  $j \neq i$ , compare the ‘individuals’  $j$  with  $i$  based on the non-domination statements.
- If any individual  $j$  is dominated, than mark the individual  $j$  as ‘dominated’.
- Repeat this procedure until  $i$  is equal to  $N$ .

In the end of this procedure, those ‘individuals’ that are not marked belong to the first non-dominated front. To find the non-dominated ‘individuals’ that fit into the second front one temporarily disregards the ‘individuals’ of the first front and reinitialize the described procedure. The procedure must be repeated until all the ‘individuals’ are classified into a level of non-domination.

Subsequently, those ‘individuals’ in the first non-dominated front (or level) receive an identical dummy fitness coefficient, and the sharing function strategy is responsible for updating these dummy fitness coefficient value emphasizing those ‘individuals’ that are isolated from the others in the same group. The sharing function strategy works with the ‘*niche principle*’, which is a biological concept. In biological terms, one can say that those ‘individuals’ in the same non-dominated front have a *habitat* with a given ‘*food resource*’. If one of these ‘individuals’ is far enough from the others, or alone, it means that it will not ‘*share his food supply*’, hence it will be better ‘*fed*’.

Once all the ‘individuals’ of the first non-dominated front receive their updated fitness value, the lowest fitness coefficient of this front can be determined. Thereafter, all the ‘individuals’ in the second front receive a dummy fitness coefficient value smaller than the smallest fitness coefficient from the previous front. Again, the sharing function is applied to this front. The process is repeated until all the ‘individuals’ in each front receive a fitness coefficient value. The sharing function concept stipulates that the ‘individuals’ of the first non-domination front have higher fitness coefficient than those of the second front, and so one. This maintains a pressure towards the Pareto-optimal solution

The fitness coefficient value assignment using the sharing function strategy can be summarize as:

- **Step 1** → Get a set of  $n_k$  ‘individuals’ in the  $k$ -th non-dominated front, each one having a dummy fitness value  $f_k$ , and pick a certain  $i$ -th individual.
- **Step 2** → Compute a normalized Euclidean distance measure in relation to another  $j$ -th individual in the same  $k$ -th non-dominated front, ( $j \neq i$ )

$$d_{ij} = \left[ \sum_{p=1}^P \left( \frac{x_p^{(i)} - x_p^{(j)}}{x_p^u - x_p^l} \right)^2 \right]^{1/2} \quad (5)$$

where  $P$  is the number of variables in the problem. The parameters  $x_p^u$  and  $x_p^l$  are the upper and the lower bounds of  $x_p$  variable.

- **Step 3** → This  $d_{ij}$  distance is compared with a pre-specified  $\sigma_{share}$  parameter and the following sharing function value is computed,

$$Sh(d_{ij}) = \begin{cases} 1 - \left( \frac{d_{ij}}{\sigma_{share}} \right)^2, & \text{if } d_{ij} \leq \sigma_{share} \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

- **Step 4** → If  $j < n_k$ , increment  $j$  and go to step 2.
- **Step 5** → If  $j = n_k$ , calculate the niche count for the  $i$ -th individual as follows:

$$m_i = \sum_{j=1}^{n_k} Sh(d_{ij}) \tag{7}$$

- **Step 6** → Degrade the dummy fitness,  $f_k$ , of the  $i$ -th individual in the  $k$ -th non-dominated front to

$$f_i' = \frac{f_k}{m_i} \tag{8}$$

- **Step 7** → If  $i < n_k$  increment  $i$  and go to step 2,
- **Step 8** → If  $i = n_k$  go to step 1, get another front and repeat the process until all the ‘individuals’ in each front have a fitness value.

At the end of this process, the roulette method (Goldeberg,1989) is applied to select those ‘individuals’ which will pass to the crossover stage. Basically, the chance to be selected grows with the increase of the fitness value. However, a high fitness coefficient does not necessarily guarantee that an individual will be selected. These selected ‘individuals’ will perpetuate their characteristics for the next population. That is the form in which information is processed in a GA concept.

### 2.4 CROSSOVER

In this process new ‘individuals’ are created by the change of features from the previously selected ‘individuals’. Frequently, this process leads to an improvement in the new population. There is a number of ways to execute the crossover (Deb and Agrawal,1995) but, in all of the methods, two ‘individuals’ are selected and in a certain position they exchange the binary strings. In this work the single-point crossover is implemented. The position where the ‘individuals’ should exchange strings is randomly chosen. An illustration of a single-point operator can be seen below. Two ‘individuals’ are chosen to create two new ‘individuals’.

1 0 0 0 1 0 1 0	1 1 0 1 1 1 0 1	<i>Individual 1</i>
1 1 1 0 0 0 1 0	1 0 1 0 0 1 1 1	<i>Individual 2</i>
1 0 0 0 1 0 1 0	1 0 1 0 0 1 1 1	<i>New Individual 1</i>
1 1 1 0 0 0 1 0	1 1 0 1 1 1 0 1	<i>New Individual 2</i>

Figure 2. Crossover procedure for the creation of new ‘individuals’.

### 2.5 MUTATION

It is a way of keeping the diversity of the population. In this process, the binary value at an aleatory position in the string suffers a random inversion in his value. If it is 1, it becomes 0 and, if it is 0, it becomes 1. Mutation promotes diversity by allowing the optimization procedure to search for ‘individuals’ in the solution space that otherwise would not be contained in the current population.

1 0 0 0 1 0 1	0	1 1 0 1 1 1 0 1	<i>Individual n</i>
1 0 0 0 1 0 1	1	1 1 0 1 1 1 0 1	<i>New Individual n</i>

Figure 3. Mutation at a certain position of the ‘individual’.

### 3 NSGA EVALUATION

An evaluation of two test cases, available in the literature, was implemented just to check the NSGA implementation. This evaluation procedure was performed to ensure that the optimized airfoils were not obtained by casual chance, but by a robust and reliable implementation. The evaluation was done with conflicting objective functions, and the minimum value of each objective function was pursued. The two test cases are described below:

(a) First test case: 
$$\begin{cases} f_1(x) = x^2 \\ f_2(x) = (x-2)^2 \end{cases} \tag{9}$$

$$(b) \text{ Second test case: } \begin{cases} f_1(x) = x \\ f_2(x, y, z, w) = g(x, y, z, w) \{1 - [x/g(x, y, z, w)]^2\} \\ g(x, y, z, w) = 1 + x + y + z + w \end{cases} \quad (10)$$

The results from this evaluation can be seen in Figs. (4) and (5). In Figure (4) one can observe that the same Pareto-front was obtained for different sizes of population and different generations, without any dispersion around the front. On the other hand, in Fig. (5), one can observe the presence of a small dispersion around the Pareto-front due to the complexity of the function tested; as well as the effect of a variation in the population size. This dispersion is related to the existence of secondary Pareto-fronts, which do not represent the best Pareto-front. One can think of these secondary Pareto-fronts as ‘local’ optimized solutions.

It can be clearly seen that there is not a single solution that satisfies the optimization but a set of possible solutions, which form the Pareto-front. It is worth mentioning that the obtained Pareto-front matches those presented in the literature (Deb *et al*, 200) for these two test cases. These promising results show that the implementation of the NSGA is satisfactory and reliable, and it can, now, be used for airfoil design, which is the major thrust of the present work.

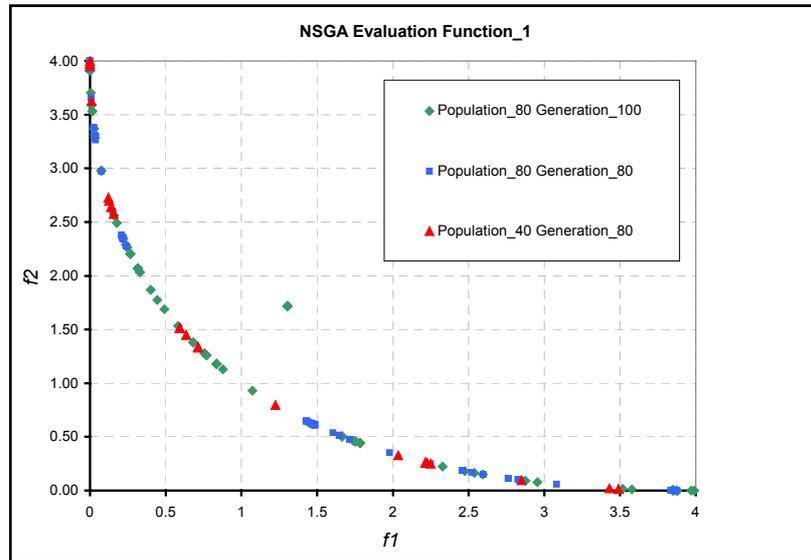


Figure 4. Pareto front for the first set of conflicting functions.

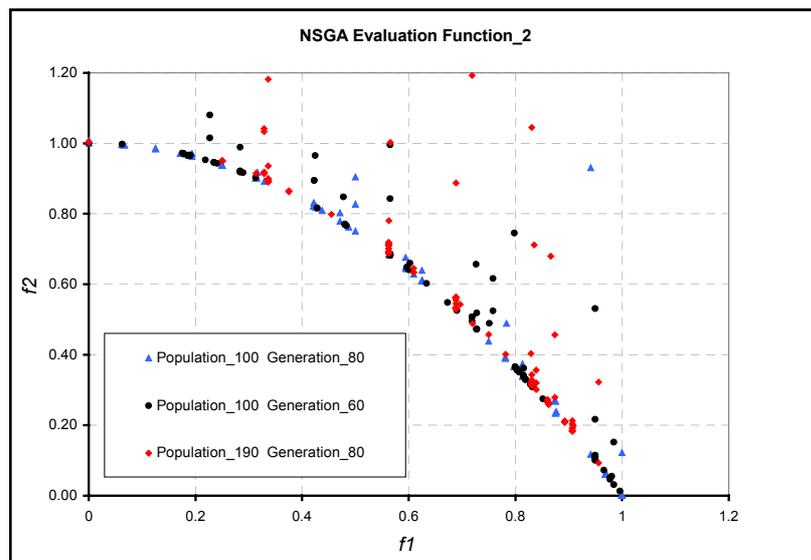


Figure 5. Pareto front for the second set of conflicting functions.

#### 4 AIRFOIL SHAPE PARAMETERIZATION

The airfoil shape parameterization was obtained with the two following polynomials:

$$t = a_1 \sqrt{x} + a_2 x + a_3 x^2 + a_4 x^3 + a_5 x^4 , \tag{11}$$

$$y_c = b_1 x + b_2 x^2 + b_3 x^3 + b_4 x^5 + b_5 x^6 . \tag{12}$$

The values of the various coefficients of the polynomials are obtained from the prescribed leading-edge radius, upper and lower crest location, curvatures, trailing-edge ordinate, thickness, direction and wedge angle, among others. The NSGA is responsible to provide the best set of these aerodynamic parameter in the sense of obtaining the lower possible drag.

#### 5 RESULTS

Initially, using a fixed population of 30 ‘individuals’, the optimization for two design conditions was performed: Mach number 0.74,  $Cl = 0.35$ , and Mach number 0.78,  $Cl = 0.35$ . Subsequently, two other optimizations were performed with an increase in the population size to 60 and 100, respectively. These analyses were performed in order to assess the sensibility with respect to the minimum population size that would produce satisfactory results.

In Fig. (6), it can be seen that the increase of the population size produced a displacement of the Pareto-Front in the direction of the lower level of drag coefficient. One can also notice that the increase of the population size from 30 to 60 ‘individuals’, produced a huge improvement, but the increase from 60 to 100 ‘individuals’ did not produce the same level of improvement. As can be seen in Fig. (6), the Pareto-front highlights a series of possible solutions, which are committed with the optimization at the design points.

During a design procedure, the computational cost is an important variable. Particularly, at the preliminary design stage, there is the need to obtain, as quickly as possible, many potentials shapes that might be used for a wing lofting. In the present work, the population size of 100 ‘individuals’ was assumed as satisfactory for the desired level of optimization. The increase of the population size, for values higher than 100 would not produce a significant difference in terms of drag coefficient, however the computational cost would be considerably higher.

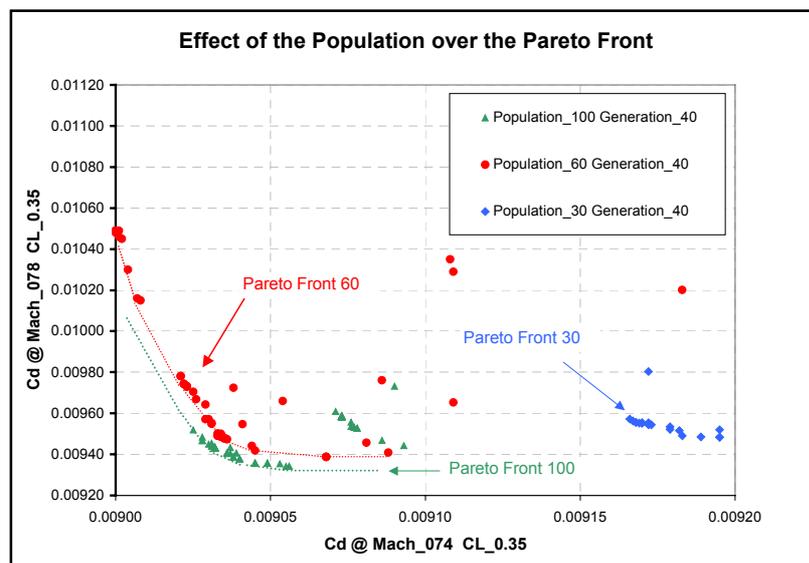


Figure 6. Effect of the population size in the Pareto front for two design conditions.

Figure (7) shows drag rise curves for three specific ‘individuals’, each one belonging to the Pareto-front of 30, 60 and 100 ‘individuals’, respectively, as shown in Fig.(6). These ‘individuals’ were chosen considering the minimum value of Cd at the two design points. One can observe that the optimized individual coming from the population size of 100 has a range in Mach number where it has a better behavior in relation to the others. Although this solution is not the best for the entire Mach number range, it is still optimized and acceptable because it is just one of many possible solutions obtained by the Pareto-front of 100 ‘individuals’. Figure (8) shows the difference in the shape for the optimized airfoils. These differences are mainly related to the increase of the population size, which is responsible for the improvement in the drag coefficient.

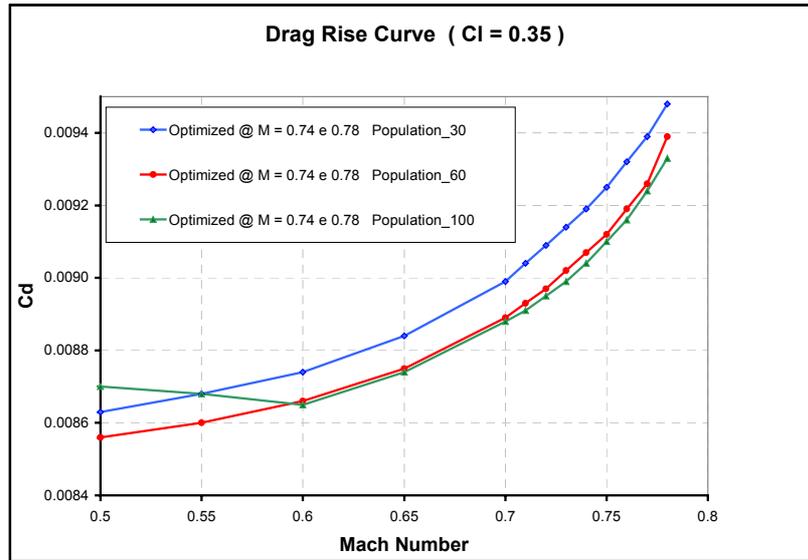


Figure 7. Drag rise curves for three specific ‘individuals’ from each of the three Pareto front.

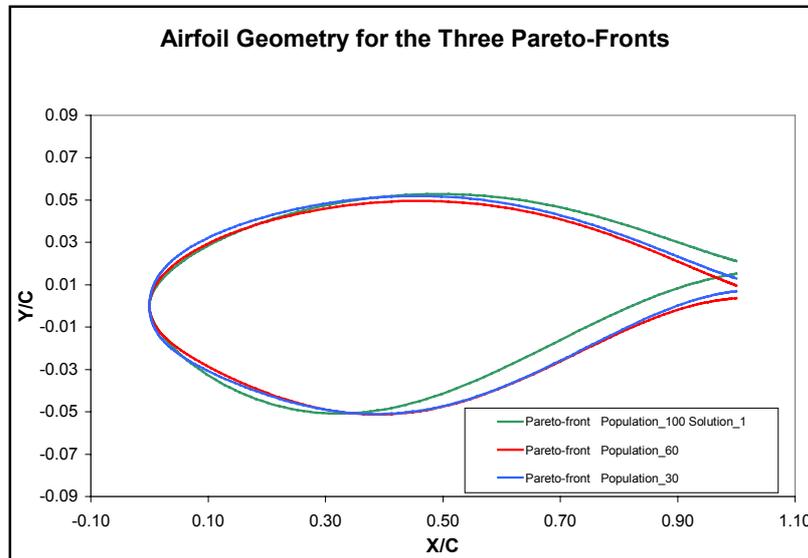


Figure 8. Geometry shape for the optimized airfoils from the Pareto-fronts of 30,60, and 100 ‘individuals’.

In Fig. (9), one can see a comparison between the drag rise curves for two solutions belonging to the same Pareto-front of 100 ‘individuals’. These two plotted curves belong to a set of 100 possible solutions and, from this set, one must choose which solution better possesses the desired aerodynamic characteristics. These ‘individuals’ from the same front are not identical to each other, and they are associated with different aerodynamic characteristics. One can also observe in Fig.(10), the shape of the two referred optimized solutions. These two shapes are quite similar; therefore, they have different aerodynamic characteristics as one can see in the drag rise curve in Fig. (9).

A single design point optimization was performed for a Mach number 0.74,  $C_l = 0.35$  and a population size of 100 ‘individuals’. This single design point result was compared, in terms of drag rise, with the prior optimization using two design points, as one can see in Fig. (11). One can observe at the Mach number range from 0.60 to 0.75, that the two-design point optimization produces a less optimized solution in terms of  $C_d$  value. The difference with respect to the single design point is about 0.5 drag counts for this Mach number range. For a Mach number lower than 0.60, this difference has a rapid increase, achieving 2.0 drag counts at Mach number 0.5. However, for a Mach number higher than 0.75, the single design point optimization reaches the divergence Mach number while the two-design point optimization still has a very reasonable level of drag coefficient.

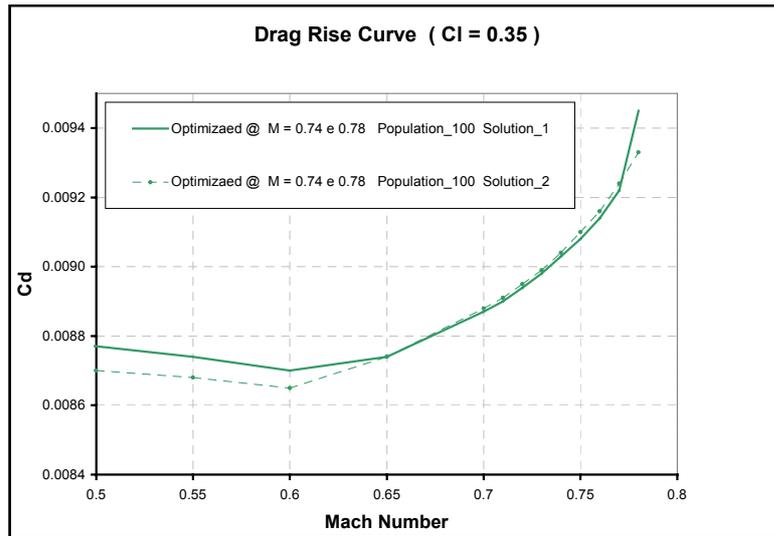


Figure 9. Comparison between two solutions of the Pareto-front from a population size of 100 'individuals'.

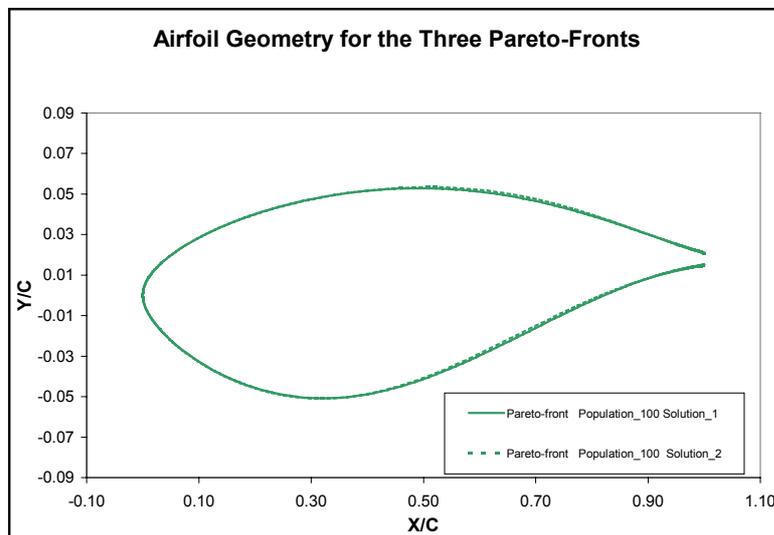


Figure 10. Geometry shape for two optimized airfoils from the Pareto-front with 100 'individuals'.

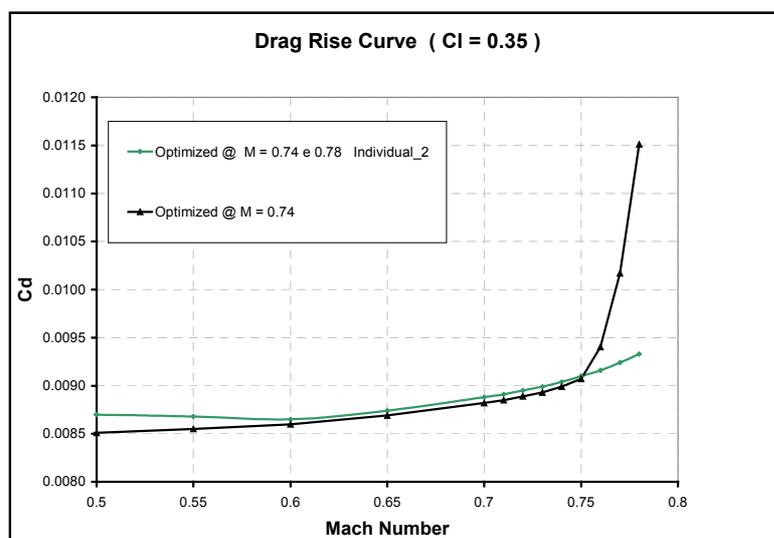


Figure 11. Drag rise curve for a single optimized airfoil and a multi design point optimized airfoil.

The presented results show that, one cannot support the two-design point as a better approach than the single design point optimization, and vice-versa. This substantiation is based on the fact that the obtained shapes are optimized at their respective design points. In the present case, the two-design point shape has a  $C_d$  value not as optimized as the single design point at Mach number of 0.74. This difference is most certainly related with the constraint imposed to the obtained shape of the two-design point airfoil to achieve a reasonable  $C_d$  value at Mach number of 0.78.

Considering solely the two-design point optimization, one can observe that there is an increase in the  $C_d$  value for Mach numbers lower than 0.60. Perhaps, the lack of a design point at a lower Mach number might be driving the Pareto-front to solutions characterized by an excessive  $C_d$  value at this range of Mach number. A way to diminish this undesired level of  $C_d$  at Mach numbers below 0.60 is to prescribe a third point of design. The logical choice for this third point should be at a Mach number below or equal to 0.60. By such an approach, one would try to guarantee that the optimization is performed for a wider range of Mach numbers. Since, the main concern of this work is related to the performance at transonic speeds, the degradation at lower Mach numbers is being accepted in benefit of the improvement achieved in the transonic range. Hence, this test with the imposition of a third design point was not performed in the present work.

Figure (12) shows the difference between the airfoil shapes obtained using the single or the two-design point optimization. The shape differences produce  $C_p$  distributions with different characteristics as one can observe in Figs. (13) and (14). At Mach number 0.60, the two-design point optimization has a suction peak higher than the single design point optimization, generating a higher value of  $C_d$  as mentioned before. But, for a Mach number of 0.78, the single design point optimization has a shock wave much stronger than the two-design point optimization, leading to drag divergence of this airfoil as indicated in Fig. (11).

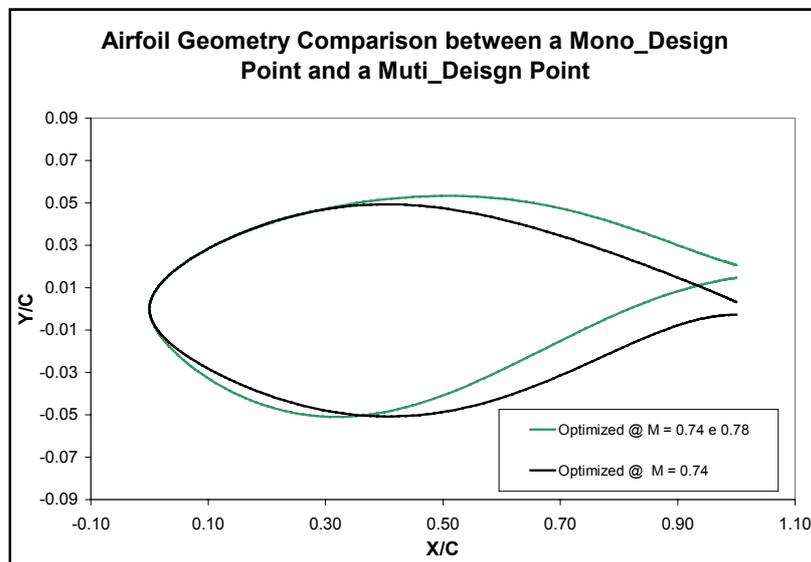


Figure 12. Geometry shape for the single design point optimized airfoil and the two-design point optimized airfoils.

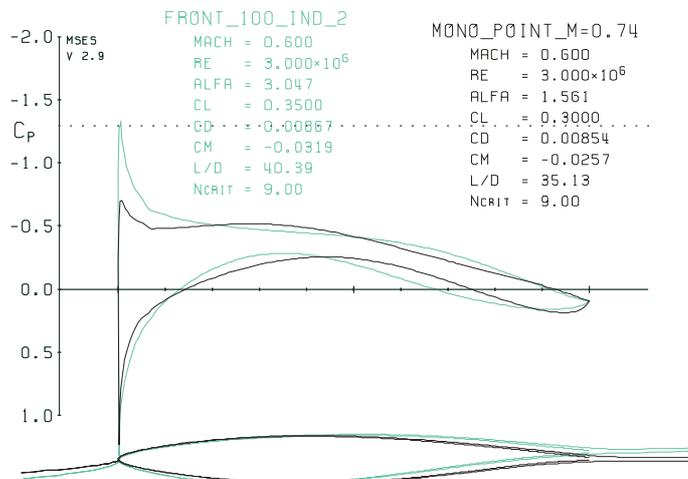


Figure 13. Comparison between the single design point optimization and the multiple design point optimization at Mach number 0.60.

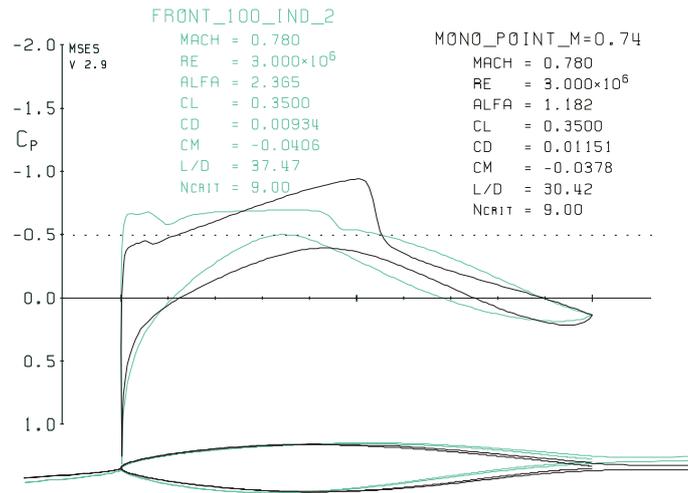


Figure 14. Comparison between the single design point optimization and the multiple design point optimization at Mach number 0.78.

## 6 CONCLUSIONS

The optimization process using a NSGA showed to be satisfactory in the search for an airfoil shape that attempts to be optimized in more than one design point. In the study cases of this work, it was shown that the two-design point optimization produced a postponement of the drag divergence and a lower level of drag coefficient for higher values of Mach number.

The lack of a design point at lower values of Mach number might have driven the obtained Pareto-front to solutions that allow larger  $C_d$  values at this range of velocity. It is believed that, for each design point (Mach number) there are some geometrical characteristics that obtain minimum achievable value of  $C_d$ . When a multi-design point optimization is being performed, one can never get the perfect aerodynamic shape for each design point. However, the optimized solutions are a trade off among the several design points. In general terms, one is responsible to take the decision of what is an acceptable aerodynamic characteristic at a certain Mach number range. In the present work, the improvement obtained at a higher Mach number was satisfactory, in spite of a lower performance at lower Mach numbers. But, sometimes one might not accept this degradation at lower velocities and, then, perform an optimization using an extra point to ensure a decrease in the  $C_d$  value for the low Mach number range.

In this work no more than two points of design optimization were studied, and the limit of design point is the computational cost that one is forced to pay. The computational costs were not so high in the present case because the authors have used a two-dimensional Euler formulation with boundary layer correction, which is the physics contained in the MSES code, in order to obtain the objective function. Therefore, as one decides to follow on to a higher fidelity code to obtain the objective function, as for example the use of a Navier-Stokes solver with advanced turbulence modeling, the cost of a design cycle can become extremely expensive.

## 7. ACKNOWLEDGEMENTS

The second author acknowledges the partial support of CNPq through the Integrated Project Research Grant N° 501200/2003-7.

## 8. REFERENCES

- Antunes, P. A., Azevedo, J.L.F., Santos, L.C., 2003, "Airfoil Shape Optimization Using Genetic Algorithms", Proceedings of the 17th ABCM International Congress of Mechanical Engineering – COBEM 2003, São Paulo, SP.
- Deb, K., 1995, "Optimization for Engineering Design: Algorithms and Examples", Prentice-Hall, Delhi, India.
- Deb, K., 1999, Kanpur Genetic Algorithms Laboratory Report Number 99002, "Multi-Objective Evolutionary Algorithms: Introducing Bias Among Pareto-Optimal Solutions".
- Deb, K., Pratap, A., Meryarivan, T., 2000, Kanpur Genetic Algorithm Laboratory Report Number 200002, "Constrained Test Problems for Multi-Objective Evolutionary Optimization".
- Drela, M., 1996, MIT Computational Aerospace Sciences Laboratory, "A User's Guide to MSES 2.95".
- Goldberg, D.E., "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison-Wesley, New York, 1989.

- Streshinsky, J.R., Ovcharenko, V.V., 1994, "Aerodynamic Design Transonic Wing Using CFD And Optimization Methods", ICAS.
- Nixon, D., 1981, "Transonic Aerodynamics", Martin Summerfield Series Edition, New York, USA.