

## AIRFOIL SHAPE OPTIMIZATION USING A CONTROLLED RANDOM SEARCH ALGORITHM

**Nelson Manzaneres-Filho, nelson@unifei.edu.br**

**Bruno Silva de Sousa, bruno\_s\_sousa@yahoo.com.br**

**Ramiro Gustavo Ramirez Camacho, rgramirez65@hotmail.com**

**Rodrigo Barbosa da Fonseca e Albuquerque, galegosigma@yahoo.com.br**

**Ariosto Bretanha Jorge, ariosto@unifei.edu.br**

Universidade Federal de Itajubá, Instituto de Engenharia Mecânica, Av. BPS 1303, C. P. 50, CEP 37500-903, Itajubá, MG, Brazil.

**Abstract.** *This paper presents a methodology for airfoil shape optimization using a global search algorithm, namely, a Controlled Random Search Algorithm (CRSA). Like genetic and differential evolution algorithms, the CRSA is a population-set based algorithm in which an initial population of prospective designs is randomly computed and better solutions are iteratively incorporated in place of worse ones, until a convergence criterion is achieved. The main advantage of the CRSA is its ease of implementation, without sacrificing too much its robustness. The airfoil shape is parameterized by two Bezier arcs of high degree representing the lower and upper surfaces. The abscissas of the control points are fixed and the ordinates are treated as design variables. Constraints are incorporated by means of a penalty scheme. A relatively low fidelity solver is adopted for testing the algorithm implementation. The solver is based on a vortex panel method with Gostelow fairing-in correction and integral methods for boundary layer calculation. Separation effects are empirically represented. Some case studies of  $C_D$  and  $C_D/C_L$  minimization are presented for testing the efficiency and robustness of the proposed methodology.*

**Keywords:** *airfoil shape optimization, controlled random search, Bezier parameterization, boundary layer*

### 1. INTRODUCTION

Aerodynamic design methods can generally be classified into two broad classes: direct and inverse (van den Dam et al., 1990). Both approaches have their own advantages and drawbacks and have been extensively applied in modern aerodynamic design.

Optimization methods constitute important tools for implementing these design approaches. In the direct approach, optimization methods are coupled with flow analysis methods in order to minimize (or maximize) a certain aerodynamic objective function by directly iterating on the geometry. The optimum pressure distribution is computed as part of the solution in conjunction with the geometry itself. In the inverse approach, the designer must furnish an adequate pressure distribution beforehand. Once this target distribution is given, a corresponding geometry can be determined by inverse methods. Here, optimization methods can be applied for optimizing the target distribution as well as for defining the required geometry.

Normally the objective functions appearing in aerodynamic design problems are strongly non-linear and multimodal. Gradient based optimization algorithms were formerly applied (Vanderplaats and Hicks, 1976) but they suffer from strong dependency of starting shapes and difficulties for escaping from local solutions. Some studies have pointed out that genetic algorithms are better suited for aerodynamic shape design in comparison with gradient based and simulated annealing algorithms (Obayashi and Tsukahara, 1997). Genetic algorithms work with a population of evolving individuals (designs). They have been applied to a variety of engineering problems during the last decades. In aerodynamic design they have been applied for solving both direct and inverse problems in aeronautics and turbomachinery (Obayashi and Takanashi, 1996; Dennis et al., 2001; Oksuz et al., 2002; Hacıoğlu and Özkol, 2005).

Besides genetic algorithms, other population-set optimization algorithms suitable for handling multimodal optimization problems were also developed. Relevant options are differential evolution algorithms (Storn and Price, 1997) and controlled random search algorithms (Price, 1977). Differential evolution has already been applied in aerodynamic shape design (Rogalsky et al., 1999). In comparison with genetic algorithms, however, much fewer papers can be found exploring these alternative algorithms in shape design problems.

Among population-set based algorithms, controlled random search algorithms (CRSA) are probably the most straightforward in terms of implementation. They were initially proposed by Price (1977) and substantially improved by Ali et al. (1997b). They have been successfully applied in a variety of real-world problems (Ali et al., 1997a) and have also been shown to be competitive with genetic and differential evolution algorithms (Ali and Törn, 2004). Some recent applications of CRSA include the conceptual optimization of axial-flow hydraulic turbines and the inverse design of isolated airfoils and turbomachinery cascades (Albuquerque et al., 2007ab, Manzaneres-Filho et al., 2005).

The present work presents a methodology for direct airfoil optimization by applying a CRSA version proposed by Ali et al. (1997b) with some modifications introduced by Manzaneres-Filho et al. (2005). The airfoil shape is parameterized by two Bezier arcs of high degree representing the lower and upper surfaces. The abscissas of the control points are fixed and the ordinates are treated as design variables (except for the points joining the arcs). Constraints are

treated by means of a penalty scheme. A relatively low fidelity flow analysis code (solver) is chosen for testing the CRSA. This solver is based on a vortex panel method (Lewis, 1991) with Gostelow fairing-in correction (Gostelow, 1984) and integral methods for boundary layer calculation (Moran, 1984).

Section 2 discusses the employed algorithm (CRSA). The airfoil shape parameterization is presented in Section 3. In Section 4, the adopted flow solver is briefly described. Some case studies of  $C_D$  and  $C_D/C_L$  minimization are presented in Section 5 for testing the efficiency and robustness of the proposed methodology. Section 6 ends the paper with some concluding remarks.

## 2. CONTROLLED RANDOM SEARCH ALGORITHMS (CRSA)

Controlled Random Search Algorithms (CRSA's) are optimization methods suitable for searching of global minimizers of a continuous real function (objective function),  $f: \mathfrak{R}^n \rightarrow \mathfrak{R}$ , not necessarily differentiable, defined on a hyper-box  $S = \{\mathbf{x} \in \mathfrak{R}^n: x_j^L \leq x_j \leq x_j^U, j = 1, \dots, n\}$ , where  $x_j^L$  and  $x_j^U$  represent, respectively, lower and upper bounds for the  $n$  coordinates of  $\mathbf{x}$ . A point  $\mathbf{x}^*$  is said to be a global minimizer of  $f$  if  $f(\mathbf{x}^*) \leq f(\mathbf{x}), \forall \mathbf{x} \in S$ . Besides the lateral constraints used in the definition of  $S$ , other types of constraints could in principle be imposed by means of a penalization scheme on objective function or by any other constraint-handling technique (Oyama et al., 2005).

CRSA's were proposed as an improvement to simple random search methods in which only the point with the lowest function value was retained in each iteration (Price, 1977). Like genetic and differential evolution algorithms, a CRSA is a population set-based algorithm that starts with an initial set  $P$  of  $N$  points randomly chosen over  $S$  and then carries out an iterative contraction process towards a global minimizer by means of purely heuristic procedures. The population size  $N$  is maintained throughout the optimization process. Unlike the other mentioned global optimization algorithms, the CRSA replaces a single point of the population (its current worst objective value point,  $\mathbf{h}$ ) by a better point  $\mathbf{p}$  in each iteration (i.e., a trial point  $\mathbf{p}$  so that  $f(\mathbf{p}) < f(\mathbf{h})$ ). Thus its implementation is more straightforward.

### 2.1. Basics of CRSA

The basic CRSA for minimization can be described in six steps as follows (adapted from Ali et al., 1997b; Ali and Törn, 2004):

1. Generate an initial population  $P$  of  $N$  random points in  $S$ :  $P = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ . Compute the function values of these points in an indexed way. Determine the worst point,  $\mathbf{h}$ , and the best point,  $\mathbf{l}$ , i.e., those points in  $P$  with the highest and the lowest function values,  $f_h$  and  $f_l$ , respectively. If a stopping criterion is already satisfied, then stop (for example, stop if  $f_h - f_l < \varepsilon$ , where  $\varepsilon$  is a given tolerance).
2. Generate a trial point  $\mathbf{p}$  for replacing the worst point,  $\mathbf{h}$ .
3. If  $\mathbf{p}$  is infeasible ( $\mathbf{p} \notin S$ ), go to step 2 (or alter  $\mathbf{p}$  to make it feasible).
4. Evaluate  $f_p = f(\mathbf{p})$ . If  $\mathbf{p}$  is unsatisfactory ( $f_p \geq f_h$ ), go to step 2.
5. Update the set  $P$  by replacing the current worst point by the trial point: ( $P \leftarrow P \cup \{\mathbf{p}\} / \{\mathbf{h}\}$ ). Find  $\mathbf{h}$  and  $f_h$  in new  $P$ . If  $f_p < f_l$ , then set  $\mathbf{p}, f_p$  as new  $\mathbf{l}, f_l$ .
6. If a stopping criterion is satisfied, stop; else go to step 2.

The two main differences among the available CRSA versions are mainly related to the generation mode of the trial point in step 2. It should be noted that all versions assume that  $N \gg n$  and it is typically suggested  $N = 10(n + 1)$ .

### 2.2. Some improved versions of CRSA

The CRSA proposed by Price (1977) was apparently the first one to appear in the format described above. The trial point generation in step 2 is carried out as follows: choose randomly  $n + 1$  distinct points from the current population  $P$ :  $\mathbf{r}_1, \dots, \mathbf{r}_{n+1}$  (forming a simplex in  $\mathfrak{R}^n$ ). Determine the centroid  $\mathbf{g}$  of the  $n$  first points  $\mathbf{r}_1, \dots, \mathbf{r}_n$ . Thence determine the trial point  $\mathbf{p}$  as the reflection of the remaining point  $\mathbf{r}_{n+1}$  about the centroid  $\mathbf{g}$ .

Ali *et al.* (1997b) have compared some modified CRSA versions. The authors have chronologically enumerated these versions as follows: the CRS1, the Price original algorithm just described above; the CRS2, also proposed by Price by making a more sophisticated use of simplexes in the trial point generation; the CRS3, also due to Price, in which a local phase was included; the CRS4, a modification of CRS2 by including local random searches around the best point using  $\beta$ -distribution variates; the CRS5, excluded from the comparisons of Ali *et al.* (1997b) since it employs a gradient based local search and the emphasis had been given in non-gradient algorithms; finally, the CRS6, proposed by Ali *et al.* (1997b), in which the local searches of CRS4 (based on  $\beta$ -distribution) are retained and the global phase uses the quadratic interpolation scheme of Palosaari *et al.* (1986). In this scheme, three distinct points of the current population  $P$  are taken: the best point,  $\mathbf{l}$ , and two other ones at random,  $\mathbf{r}_2$  and  $\mathbf{r}_3$ . By varying  $j$  from 1 to  $n$ , quadratic interpolations are constructed using the corresponding coordinates of those three points,  $l_j, r_{2j}$  and  $r_{3j}$ . The coordinate  $p_j$

of the trial point is equated to the *vertices* point of the quadratic. The heuristic idea behind this scheme is to consider as promising any region around the best point and to promote a global search for better points in these regions by means of coordinate-wise interpolations. Ali et al (1997b) reported very good test results of CRS6 in comparison with other CRSA versions. Those authors have also verified that the quadratic interpolation scheme is more beneficial than the local searches based on  $\beta$ -distribution variates. The version of CRS6 without local searches is named CRSI (Ali and Törn, 2004)

As pointed out by Albuquerque et al. (2007 b), the interpolation scheme used in CRSI/CRS6 has basically two features: (i) when the vertices is a minimum, the search is more local, more intensified, although even when this occurs the trial point may lies far from the current best point (for instance, this can occur when the interpolation is ill-conditioned); (ii) when the vertices is a maximum, the search is more global, more diversified. These features were not quite clear in the work of Ali et al. (1997 b), and it seems that they are both responsible for the relative good performance of the algorithm.

Albuquerque et al. (2007 b) have proposed and compared two other improved versions of CRSA: (1) an improved CRSI; (2) and a variability based reflection algorithm with improved exploratory control, named CRS-VBR\_E. The interested reader can find more detailed information about these algorithms in the cited reference. Here, only some basic information concerning the aims of the present work is given.

The improved CRSI introduces two slight modifications in the original CRSI: (i) an *a priori* test of ill-conditionality of the quadratic interpolation is performed and (ii) when the vertices of the quadratic lies out of the corresponding search interval, the trial point coordinate is equated to the adjacent coordinate limit (forcing feasibility scheme).

The CRS-VBR\_E is an extension of a previously proposed CRSA version named CRS-VBR (Manzanares-Filho et al., 2005). The CRS-VBR (Controlled Random Search using Variability Based Reflections), makes a selective use of the quadratic interpolations of CRSI and takes into account the function variability around the current best point. Like in CRSI, three distinct points are taken from the current population  $P$ : the best point,  $\mathbf{I}$ , and two other ones at random. Then a mean function value and a local variability measure around the best point are calculated and used for evaluating the trial point.

The CRS-VBR has performed better than either CRS6 or CRSI in some tests of inverse airfoil shape design (Manzanares-Filho, 2005). Curiously, when applied to some traditional benchmark test problems, the CRS-VBR has performed worse than CRSI (Albuquerque et al., 2006). Further, both algorithms have shown a relatively poor performance in solving some benchmark problems categorized as difficult or moderately difficult in the specialized literature (Ali and Törn, 2004). These issues have motivated the introduction of improvements in CRS-VBR (leading to CRS-VBR\_E algorithm). Basically, the improvements aim to satisfy the following requests: (i) a better control of the quadratic interpolations usage of CRSI; (ii) a more selective application of the variability based reflections of CRS-VBR in order to provide a more exploratory searching in the initial phases of the optimization process and reduce the chance of premature population contractions; (iii) an improved feasibility forcing scheme based on the function variability.

The proposed algorithms by Albuquerque et al. (2007 b) were comparatively tested in some benchmark problems in terms of robustness (rate of success) and computational effort (number of function evaluations). It was verified that CRS-VBR\_E generally outperforms CRSI in terms of robustness but is outperformed in terms of computational effort.

The algorithms were also comparatively tested in an inverse airfoil shape design problem. In this case, both algorithms were considered well succeeded in all tests, but CRS-VBR\_E has greatly outperformed CRSI, mainly in terms of computational effort. This comparative trend is specially verified when only normal reflections were used in the CRS-VBR\_E algorithm. In this case, CRS-VBR\_E essentially becomes the earlier CRS-VBR version. These results corroborated those results previously obtained by Manzanares-Filho et al. (2005), by indicating that the CRS-VBR algorithm is a very convenient choice for aerodynamic design applications. Thus, only the CRS-VBR was chosen for the direct airfoil optimization approach proposed in this work.

### 2.3. Constraint handling

Constraints can be introduced in CRSA's by means of a penalty scheme. Suppose that  $g_i \leq 0$ ,  $i=1, \dots, NI$ , and  $h_i = 0$ ,  $i=1, \dots, NE$ , represent inequality and equality constraints, respectively. A modified (penalized) objective function  $f_{pen}$  is formed in terms of the original objective function  $f$  as:

$$f_{pen} = f \left( 1 + \sum_{i=1}^{NI} G_i \max(0, g_i) + \sum_{i=1}^{NE} H_i |h_i| \right) \quad (1)$$

where  $G_i$  and  $H_i$  represent positive penalty factors to be defined by the user. This choice is problem dependent and should be made with due care: a too small factor can accelerate the algorithm, but may not be effective in promoting constraint satisfaction; on the other hand, a too large factor may lead to a loss of information about the original objective function and a hampering of the algorithm convergence.

It is important to mention that the penalty scheme in Eq. (1) is suitable for treating non-linear constraints that depends on results obtained by the analysis code *a posteriori*. *A priori* geometrical constraints directly affecting the shape design parameters are intrinsically treated by the step 3 of the basic CRSA algorithm (see section 2.1).

### 3. AIRFOIL SHAPE PARAMETERIZATION

A geometric parameterization similar to that described by Hacıoğlu and Özkol (2005) is employed in this work. The lower and upper surfaces of the airfoil are represented by Bezier curves of high degree. A Bezier curve of degree  $n_B$  in the  $(x, y)$  plane is defined by a set of  $n_B + 1$  control points  $\mathbf{P}_i = (P_{x_i}, P_{y_i})$ . A point  $\mathbf{r}(x, y)$  of the curve is given by:

$$\mathbf{r}(t) = \sum_{i=0}^{n_B} \frac{n_B!}{i!(n_B-i)!} t^i (1-t)^{n_B-i} \mathbf{P}_i \quad (2)$$

where  $t$  represents the curve parameter varying between 0 and 1. The first control point coincides the initial curve point ( $t = 0$ ); the last control point coincides the final curve point ( $t = 1$ ).

The first and last control points of both curves are fixed at the airfoil leading and trailing edges respectively. The abscissas of the remaining control points are also fixed and only their ordinates are treated as design variables. Being  $n_{BE}$  and  $n_{BI}$  the degrees of upper (extrados) and lower (intrados) curves, respectively, it results a number of design variables  $n = n_{BE} + n_{BI} - 2$ . Figure 1 shows an approximation of airfoil NACA 65<sub>1</sub>-412 by the described Bezier curves with  $n_{BE} = 6$  and  $n_{BI} = 6$  and equally spaced abscissas. It is also shown a discretization of the airfoil contour using 120 panels. This configuration will be used as a reference shape for the test examples to be presented later. So, a total of  $n = 10$  design variables should be determined in this case.

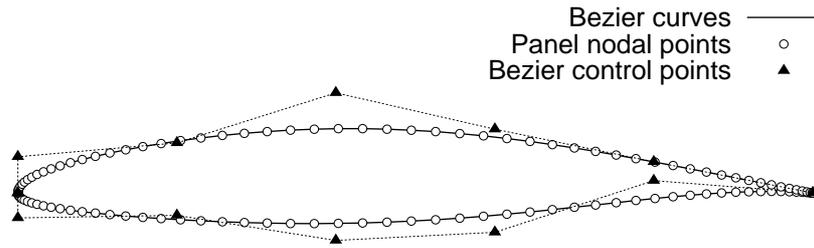


Figure 1. Approximation of airfoil NACA 65<sub>1</sub>-412 by Bezier curves

### 4. THE ADOPTED FLOW ANALYSIS CODE (SOLVER)

Population-set based algorithms like genetic or CRSA normally demand a great number of function evaluations in order to achieve a near global solution. On the other hand, an aerodynamic shape design normally requires a high fidelity flow solver (Navier-Stokes based) for obtaining a sufficiently precise geometric representation. Such kind of solver, however, is computationally expensive and is not adequate for preliminary tests of new optimization methodologies like that proposed in this paper. For this reason, a relatively low fidelity solver was adopted for testing the CRS-VBR algorithm.

This solver is based on a low order vortex panel method with curvature correction, basically described by Lewis (1991). A first approximation for viscous effects is introduced by means of a fairing-in correction of the potential pressure distribution as proposed by Gostelow (1984). This corrected pressure distribution is used for boundary layer calculations for the upper and lower airfoil surfaces by means of integral methods (Moran, 1984). The laminar portions are calculated by the method of Thwaites and the turbulent portions by the method of Head. Laminar-to-turbulent transition can be forced or prescribed by the Michel criterion.

The drag coefficient is calculated by summing up the momentum defect contributions of the upper and lower airfoil surfaces. Each contribution is calculated by the Squire-Young formula:

$$C_D = 2\theta_{te} \left( \frac{U_{e,te}}{U_\infty} \right)^{\frac{H_{te}+5}{2}} \quad (3)$$

where  $\theta_{te}$  and  $H_{te}$  are the momentum thickness and the shape factor at the trailing edge, respectively,  $U_{e,te}$  is the potential velocity at the trailing edge (corrected) and  $U_\infty$  is the far field velocity.

Separation effects are represented by an empirical approach due to Speidel, as described by Schlichting (1959). In the examples to be presented, however, the boundary layers will be constrained to not separate and these effects will eventually disappear at convergence of the optimization algorithm.

The lift coefficient  $C_L$  is calculated by integrating the corrected pressure distribution ( $C_{L,no\ wake}$ ) and thence applying a wake correction analogous to that reported by Smetana (1992):

$$C_L = C_{L,no\ wake} \left( 1 - 3T \sqrt{C_D} \right) \quad (4)$$

where  $T$  represent a thickness function measure calculated by a clockwise integral of the corrected pressure coefficient along the airfoil surface ( $L$  is the chord length):

$$T = -\frac{1}{4L} \oint C_p ds \quad (5)$$

This quantity approximates the maximum thickness to chord ratio for slightly cambered airfoils at low incidences (van Egmond, 1990). It is useful in optimization of target pressure distributions for inverse methods. It was used here instead of the real thickness for uniformity reasons and also for imposing thickness type constraints *a posteriori*.

## 5. TEST EXAMPLES

The airfoil NACA 65<sub>1</sub>-412 is used as reference for the test examples of this paper. The range of design variables is defined by perturbations of the control point ordinates shown in Fig. 1 (except for those representing the leading and trailing edges that will remain fixed). Although not strictly necessary, this approach is convenient here since the low fidelity solver is not able to tackle all required flow situations with confidence. The ordinates of the two control points close to the leading edge are perturbed in  $\pm 10\%$ . In this way, the leading edge radius is not allowed to vary too much. The remaining control point ordinates of lower curve are perturbed in  $\pm 40\%$ . The remaining control point ordinates of upper curve are perturbed in the range between  $-40\%$  and  $+140\%$ . An initial population with 110 airfoils is randomly set up by using a uniform probability distribution. In this way, a great variety of airfoils is allowed in the initial population without extrapolating too much the flow solver capabilities.

All tests were carried out with an angle of attack equal to zero and a Reynolds number equal to  $3 \times 10^6$ . The transition was forced at  $X = 0.05$  at both upper and lower surfaces ( $X$  is the airfoil abscissa normalized with the chord length). Although the reference airfoil is laminar and a natural transition could be selected, the fixed transition near the leading edge allows a more stable optimization process and was adopted here with the main purpose of testing the CRS-VBR algorithm. The aerodynamic coefficients obtained for the reference airfoil were  $C_L = 0.327$ ,  $C_D = 0.00936$ ,  $C_D / C_L = 0.0286$ . The corresponding thickness measure results  $T = 0.0860$ . The following constraints are imposed: (1) the lift coefficient must not be lower than 0.330; (2) the abscissa corresponding to the upper surface minimum  $C_p$  must not be lower than 0.4; (3) the maximum pressure slope  $dC_p/dX$  at the upper surface in the range  $0.05 \leq X \leq 0.95$  must not be greater than 2.5; (4) the thickness measure  $T$  must not be lower than 0.0860 neither higher than 0.0900; (5) The separation points are not allowed to occur for  $X < 0.95$ . Constraint (1) is self-evident. Constraints (2) and (3) help to avoid an excess of inflection in the resultant pressure distribution. Constraints (4) help to control the airfoil thickness. For low incidences, separation is normally avoided along the whole chord and constraints (5) are not activated at convergence. A unique penalty factor equal to 500 was applied to all constraints.

Two test examples are presented here: one for the minimization of  $C_D$  and other for the minimization of  $C_D / C_L$ . In both of these examples, a set of 5 independent runs of the CRS algorithm was carried out. A run is stopped when one of following criteria is satisfied: the number of function evaluations exceeds 2500; the maximum objective function difference in the population becomes lower than  $10^{-4}$ . The best test results are presented in Table 1.

Table 1. Best test results in 5 independent runs of the CRS-VBR algorithm.  
 $\alpha = 0^\circ$ ;  $Re = 3 \times 10^6$ .

Airfoil	$C_L$	$C_D$	$C_D / C_L$	Average number of FE
NACA 65 <sub>1</sub> -412	0.327	0.00936	0.0286	-----
Test 1 ( $C_D$ min)	0.330	<b>0.00933</b>	0.0283	1605
Test 2 ( $C_D / C_L$ min)	0.502	0.00979	<b>0.0195</b>	2500

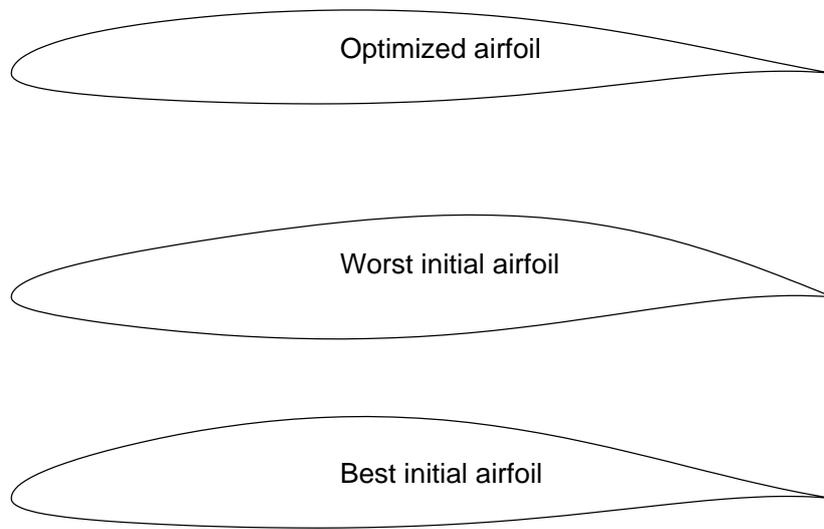


Figure 2. Shape of optimized airfoil; best and worst airfoils of initial population (best run of  $C_D$  minimization)

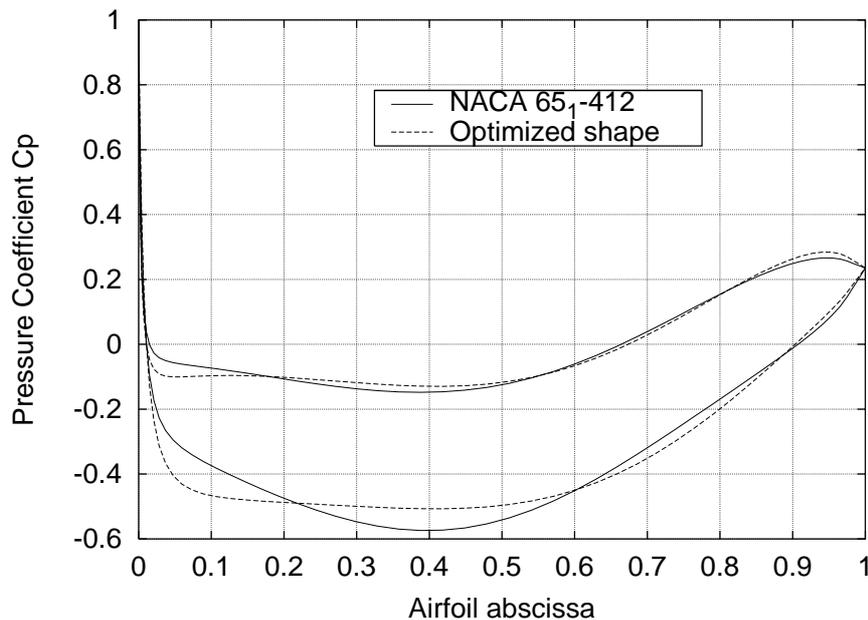


Figure 3. Comparison of pressure distributions of reference and best obtained airfoil ( $C_D$  minimization)

The best aerodynamic coefficients obtained for the  $C_D$  minimization were  $C_L = 0.330$  and  $C_D = 0.00933$  in a run with 2137 function evaluations (FE). Note, however, that a  $C_D$  with three significant figures was obtained in this run with less than 600 function evaluations. Most of the run evolution is thus spent in contracting the whole population. The average number of function evaluations of all 5 runs was 1605 for an average value of the lowest population drag equal to 0.00935. Figures 2 to 4 show some results of the  $C_D$  minimization. Figure 2 shows a comparison of the optimized airfoil and the worst and best airfoils of the initial population for the best run. Figure 3 compares the pressure distribution of the reference and optimized airfoils. Figure 4 compares the shapes of the reference and optimized airfoils with an ordinate scale-up for better visualization. It should be noted that the CRS-VBR algorithm is able to localize an optimum airfoil from a much diversified initial population. In comparison with the reference airfoil, the optimized shape has an upper surface slightly flatter which leads to a substantially flatter pressure distribution.

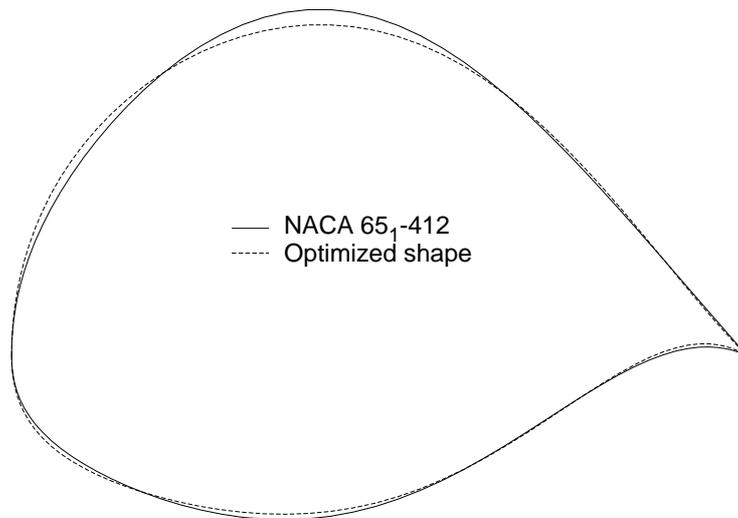


Figure 4. Comparison of shapes of reference and best obtained airfoil ( $C_D$  minimization).  
(Ordinates scaled-up for better visualization)

The best aerodynamic coefficients obtained for the  $C_D/C_L$  minimization were  $C_L = 0.502$ ,  $C_D = 0.00979$ ,  $C_D/C_L = 0.0195$  in a run with 2500 function evaluations (best run). Note, however, that a  $C_D/C_L$  with three significant figures was obtained in this run with less than 770 function evaluations. In this case, the criterion of population convergence was not attained in any of the runs. Thus, the average number of function evaluations of all 5 runs results in 2500 for an average value of the lowest population drag/lift ratio equal to 0.0201. Figures 5 to 7 show some results of the  $C_D/C_L$  minimization. Again, it should be noted that the CRS-VBR algorithm is able to localize an optimum airfoil from a much diversified initial population (Fig. 5). In comparison with the reference airfoil, the optimized shape produces now a substantially greater lift. The minimum pressure coefficient occurs for an abscissa greater than 0.4 which indicates that the constraint (2) was not activated. The pressure distribution on the upper surface is less flat than that of reference airfoil and exhibits a clear inflection at  $X \cong 0.1$ . This occurs since the optimized airfoil is substantially more cambered for greater abscissas (Fig. 7). Perhaps in this case the designer should be careful about the fidelity of the results keeping in mind the weaknesses of the adopted flow solver.

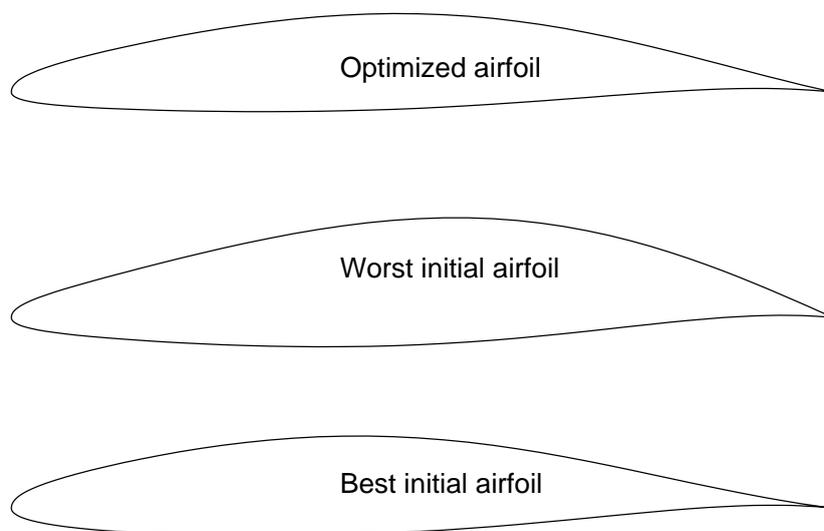


Figure 5. Shape of optimized airfoil; best and worst airfoils of initial population (best run of  $C_D/C_L$  minimization)

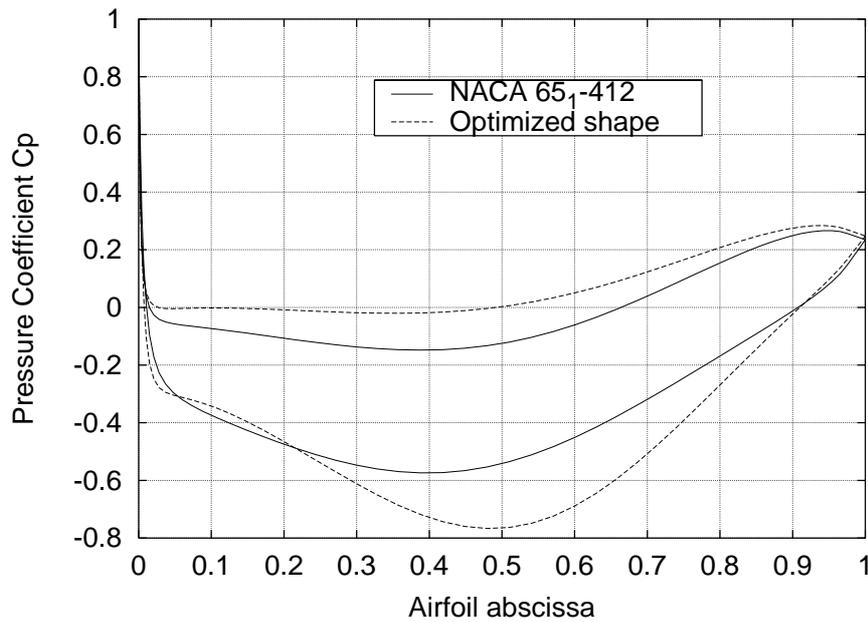


Figure 6. Comparison of pressure distributions of reference and best obtained airfoil ( $C_D/C_L$  minimization)

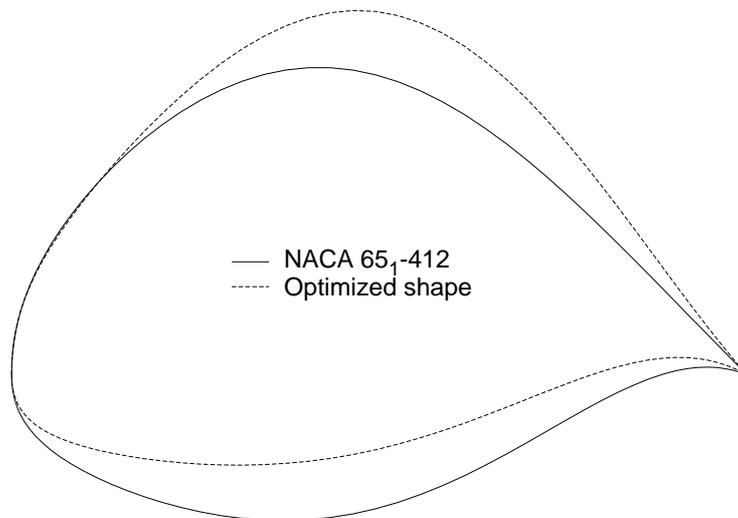


Figure 7. Comparison of shapes of reference and best obtained airfoil ( $C_D/C_L$  minimization)  
(Ordinates scaled-up for better visualization)

Although the fidelity of the results may be questionable, the relevant issue of this paper is to demonstrate the capabilities of the controlled random search algorithm (CRS-VBR) in dealing with direct airfoil shape optimization. For obtaining a satisfactory solution, a number of function evaluations smaller than 1000 was required in the test problems. For 10 design variables, this number can be considered normal when applying population-set based algorithms. Thus, two design options can be envisaged when applying the CRS-VBR algorithm. First, the designer may accept the low fidelity results as a guide for *a posteriori* refinement or validation using a few flow analyses with a higher fidelity solver. This option is attractive in terms of computation cost: using the low fidelity solver, a run with 10,000 function evaluations takes less than 5 minutes on a Pentium IV computer with 3.0 GHz and 1 Mb RAM. The other option is to couple CRS-VBR with a higher fidelity solver *a priori*. Naturally, this option is more robust but it is also very expensive in computational terms. In this case, accelerating schemes should normally be applied to the algorithm in order to reduce the number of effective function evaluations required for attaining satisfactory solutions.

## 6. CONCLUDING REMARKS

A methodology for direct airfoil shape optimization was presented. The optimization method is based on a population-set based algorithm, namely, a controlled random search algorithm (CRS-VBR).

Two test examples were made by coupling the CRS-VBR algorithm with a low fidelity flow solver. These examples have shown that the CRS-VBR algorithm can be considered as suitable for airfoil shape optimization purposes. When using this solver, the number of function evaluations for obtaining optimal solutions can be considered satisfactory (less than 1000 with 10 geometrical design variables).

The eventual weaknesses of the flow solver can be removed by using a higher fidelity flow solver (based on Navier-Stokes equations). In this case, however, is advisable to implement accelerating mechanisms in order to reduce the number of effective function evaluations required by the CRS-VBR algorithm.

## 7. ACKNOWLEDGEMENTS

During this work, the second and the fourth authors received financial supports from CAPES — Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, Brazilian Government Agency.

## 8. REFERENCES

- Albuquerque, R. B. F., Manzanares-Filho, N. and Oliveira, W., 2006, "A Study of Controlled Random Search Algorithms with Application to Conceptual Design Optimization of Axial-Flow Hydraulic Turbines", in Proceedings of XXVII CILAMCE, Belém-PA-Brazil, paper CIL09-508.
- Albuquerque, R.B.F., Manzanares-Filho, N. and Oliveira, W., 2007a, "Conceptual Optimization of Axial-Flow Hydraulic Turbines with Non-Free Vortex Design", Proceedings of the Institution of Mechanical Engineers, Vol. 221, part A: Journal of Power and Energy, pp. 713-725.
- Albuquerque, R.B.F., Manzanares-Filho, N. and Souza, B.S., 2007b, "A Comparative Study of Controlled Random Search Algorithms with Application to Inverse Airfoil Design", CMNE/CILAMCE 2007, Porto, Portugal.
- Ali, M. M., Storey, C. and Törn, A., 1997a, "Application of Some Stochastic Global Optimization Algorithms to Practical Problems", Journal of Optimization Theory and Applications, Vol. 95, pp. 545-563.
- Ali, M. M., Törn, A. and Viitanen, S., 1997b, "A Numerical Comparison of Some Modified Controlled Random Search Algorithms", Journal of Global Optimization, Vol. 11, pp. 377-385.
- Ali, M. M. and Törn, A., 2004, "Population Set-Based Global Optimization Algorithms: Some Modifications and Numerical Studies", Computers & Operations Research, Vol. 31, pp. 1703-1725.
- Dennis, B. H., Dulikravich, G. S. and Han, Z., 2001, "Optimization of Turbomachinery Airfoils with a Genetic/Sequential-Quadratic-Programming Algorithm", Journal of Propulsion and Power, Vol. 17, No. 5, pp. 1123-1128.
- Gostelow, J.P., 1984, "Cascade Aerodynamics", Pergamon Press, London, 270 p.
- Hacıoğlu, A. and Özkol, İ., 2005, "Inverse Airfoil Design by Using an Accelerated Genetic Algorithm via Distribution Strategies", Inverse Problems in Science and Engineering Vol. 13, No. 6, pp. 563-579.
- Lewis, R. I., 1991, "Vortex Element Methods for Fluid Dynamic Analysis of Engineering Systems", Cambridge University Press, 588 p.
- Manzanares-Filho, N., Moino, C. A. and Jorge, A. B. 2005, "An Improved Controlled Random Search Algorithm for Inverse Airfoil Cascade Design", 6th World Congress of Structural and Multidisciplinary Optimization (WCSMO-6), paper n. 4451.
- Moran, J., 1984, "An Introduction to Theoretical and Computational Aerodynamics", John Wiley & Sons, New York, 464 p.
- Obayashi, S. and Takanashi, S., 1996, "Genetic Optimization of Target Pressure Distributions for Inverse Design Methods", AIAA Journal, Vol. 34, No.5, pp. 881-886.
- Obayashi, S. and Tsukahara, T., 1997, "Comparison of Optimization Algorithms for Aerodynamic Shape Design", AIAA Journal, Vol. 35, No.8, pp. 1413-1415.
- Oyama, A., Fujii, K., Shimoyama, K. and Liou, M.-S., 2005, "Pareto-Optimality-Based Constraint-Handling Technique and Its Application to Compressor Design", 17th AIAA CFD Conference, paper AIAA2005-4983.
- Oksuz, O., Akmandor, I.S. and Kavsaoglu, S., 2002, "Aerodynamic Optimization of Turbomachinery Cascades Using Euler/Boundary-Layer Coupled Genetic Algorithms", Journal of Propulsion and Power, Vol. 18, No. 3, pp. 652-657.
- Palosaari, S. M., Parviainen, S., Hiironen, J., Reunanen, J. and Neittaanmaki, P., 1986, "A Random Search Algorithm for Constrained Global Optimization", Acta Polytechnica Scandinavica, Chemical Technology at Metallurgy Series, No. 172, Helsinki, pp. 1-45.
- Price, W. L., 1977, "A Controlled Random Search Procedure for Global Optimisation", The Computer Journal, Vol. 20, pp. 367-370.

- Rogalsky, T., Derksen, R. and Kocabiyic, S., 1999, "An Aerodynamic Design Technique for Optimizing Fan Blade Spacing", Proceedings of the 7th Annual Conference of the Computational Fluid Dynamics Society of Canada, Halifax, , May 30 - June 1, pp. 2-29 - 2-34.
- Schlichting, H, 1959, "Application of Boundary-Layer Theory in Turbomachinery", ASME Journal of Basic Engineering, pp. 543-551.
- Smetana, F. O., 1992, "Introductory Aerodynamics and Hydrodynamics of Wing and Bodies: A Software-Based Approach", AIAA Educational Series, Raleigh, North Carolina.
- Storn, R. and Price, K., 1997, "Differential Evolution—A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces", Journal of Global Optimization, Vol. 11, pp. 341–359.
- van Egmond, J.A., 1990, "Numerical Optimization of Target Pressure Distributions for Subsonic and Transonic Airfoil Design", Computational Methods for Aerodynamic Design (Inverse) and Optimization, AGARD CP 463, Ref. 17.
- van den Dam, R.F., van Egmond, J.A. and Slooff, J. W., 1990, "Optimization of Target Pressure Distributions", Special Course on Inverse Method for Airfoil Design for Aeronautical and Turbomachinery Applications, AGARD Rept. 780, Ref. 3.
- Vanderplaats, G. N. and Hicks, R. M., 1976, "Numerical Airfoil Optimization Using a Reduced Number of Design Coordinates", NASA TM X-73151.

## **9. RESPONSIBILITY NOTICE**

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