COBEM-2017-0248
GLOBAL OPTIMIZATION BASED ON METAMODELS USING RADIAL BASIS FUNCTION WITH APPLICATION TO AIRCRAFT WINGS

Nelson Jose Diaz Gautier
Universidade Federal de Itajubá-UNIFEI, Mechanical Engeenering Institute, Itajubá, Brazil.
Universidad Nacional Experimental de la Fuerza Armada-UNEFA, Engeenering School, San Cristobal, Venezuela.
nelsondiaz@unifei.edu.br

Nelson Manzanares Filho
Edna R. Da Silva
Universidade Federal de Itajubá-UNIFEI, Mechanical Engeenering Institute, Itajubá, Brazil.
ednaunifei@yahoo.com.br

Abstract. *This paper discusses the viability of using a metamodeling technique in conjunction with a Controlled Random Search Algorithm (CRSA) applied to aerodynamic wing design. CRSA is a stochastic algorithm capable of performing global optimization tasks efficiently. The proposed metamodel technique is based on the iterative construction of response surfaces and application of a robust search pattern. Some case studies are presented for testing the efficiency and robustness of the proposed methodology. These cases include the Dixon-Szegö test functions and an aerodynamic aircraft wing design application. Since the main objective of this paper is of prospective nature, a low-fidelity flow computation code solver for aerodynamic wing design was employed with bases on a first order 3D panel method and a boundary layer model. The main objective is the minimization of aerodynamic coefficient relations* (C_D/C_L) *and* ($1/C_L$) *with a view to future use of high fidelity codes based on CFD.*

Keywords: *aerodynamic, wing, optimization, CRSA, CORS.*

1. INTRODUCTION

A classic problem on the 3D aerodynamic analysis consists in calculation of forces and moments due to air flow around aircraft wings, Among the available computational tools, a relatively fast option is the use of panel methods (Kazt and Plotkin, 1991) with boundary layer models (Moran, 1984). It will be adopted in this work for testing an optimization methodology based on metamodels in order to accelerate optimization problems in aeronautical applications using CFD in the near future. The acceleration method to be described here is named Constrained Optimization using Response Surfaces (CORS - Regis and Shoemaker, 2005). The metamodel construction (response surface) is based on radial basis functions. The optimization engine used on metamodels is named Controlled Random Search Algorithm (CRSA - Silva, 2009).

2. THE CORS STRATEGY FOR GLOBAL OPTIMIZATION USING METAMODELS

The strategy is meant for finding a global minimum of a box-constrained continuous function f: $D \rightarrow \Re$, where $D = \{\boldsymbol{x} \in \Re^d$: $x_j^L \leq x_j \leq x_j^U$, $j = 1, ..., d\}$; x_j^L and x_j^U are lower and upper bounds for the d coordinates of $\boldsymbol{x}$ respectively. A point $\boldsymbol{x}^*$ is said to be a global minimum of f if $f(\boldsymbol{x}^*) \leq f(\boldsymbol{x})$, $\forall \boldsymbol{x} \in D$. The focus is on problems where f is a black box function that is expensive to evaluate. Thus it is important to find a point $\boldsymbol{x}' \in D$ such that $f(\boldsymbol{x}')$ is close to $f(\boldsymbol{x}^*)$ using only a relatively small number of function evaluations (Regis and Shoemaker, 2005).

The CORS general algorithm is given below (Regis and Shoemaker, 2005).

Step 1 (*Select initial points*).

Set i: = 1 and select a finite initial set of points $S_1 = (\boldsymbol{x}_1, . . . , \boldsymbol{x}_k) \subseteq D$ for costly function evaluation.

Step 2 (*Do costly function evaluation*).

Evaluate the function f at the points in S_1 and update the best function value encountered at every function evaluation.

Step 3 (*Iterate*).

While the termination condition is not satisfied, do

Step 3.1 (*Fit or update response surface*).

Fit or update a response surface model f_i' using the data points $D_i = \{(\boldsymbol{x}, f(x)\text{: } \boldsymbol{x} \in S_i\}$

Step 3.2 (*Select candidate point: auxiliary problem*).

Select the candidate point $\boldsymbol{x}_{k+i}$ for function evaluation to be a point $\boldsymbol{x}$ that solves the following constrained optimization problem:

Minimize $f'(x)$

Subject to

$$\left\|\boldsymbol{x}-\boldsymbol{x}_j\right\| \geq \beta_i \Delta_i, \quad j=1,\ldots,k+i-1$$
$$x \in D \tag{1}$$

Where

$$\Delta = \max_{\boldsymbol{x}' \in D} \min_{1 \leq j \leq k+i-1} \left\|\boldsymbol{x}'-\boldsymbol{x}_j\right\| \tag{2}$$

And $0 \leq \beta_i \leq 1$ is a parameter to be set by the user.

Step 3.3 (*Do costly function evaluation*).

Evaluate the function f at $\boldsymbol{x}_{k+i}$ and update the best function value encountered so far.

Step 4.4 (*Update information*).

$S_{i+1} := S_i \cup \{ \boldsymbol{x}_{k+i}\}$; $D_{i+1} := D_i \cup \{(\boldsymbol{x}_{k+i}, f(\boldsymbol{x}_{k+i}))\}$

Reset $i := i + 1$.

End.

The focus is on problems where f is a black box function that is expensive to evaluate. Thus it is important to find a point $\boldsymbol{x}' \in D$ such that $f(\boldsymbol{x}')$ is close to $f(\boldsymbol{x}^*)$ using only a relatively small number of function evaluations (Regis and Shoemaker, 2005).

Above, k is the number of initial evaluations points and i is the iteration counter. A search pattern $\langle\beta_1, \beta_2, \ldots, \beta_N\rangle$ is chosen with $1 \geq \beta_1 \geq \beta_2 \geq \ldots \geq \beta_N$ and applied in cycles of N iterations such that $\beta_i = \beta_{i+N}$, (Silva et al., 2009). For the solution of the auxiliary problem, we propose here the use of a Controlled Random Search Algorithm – CRSA (Ali et al., 1997; Manzanares-Filho et al., 2005).

3. THE CORS COMPUTATIONAL EXPERIMENTS

The proposed CORS implementation was first tested on the Dixon-Szegö test functions (Dixon and Szegö, 1991). The CRSA was applied 10 times. A population equal to $10(d+1)$ is adopted (d is the problem dimension). The computational experimental procedure we propose here is described by Silva (2009). Table 1 shows the obtained results using three different radial basis function models with the search pattern $\langle 0.95,0.50,0.25,0.005,0.0005,0.00\rangle$, N=6. The value of shape parameter c used for each function is also included. CORS^{PP} denotes the implementation of the present paper.

Analyzing the results in Table 1, we see that the choice of radial basis function and shape parameter has a large influence on results. Inverse multiquadric, with or without polynomial, produced better results than thin plate spline with polynomials. Using the average values of CORS^{PP*} and CRSA, we see that the metamodeling strategy was able to reduce the number of function evaluations substantially. It is important to report that the rate of success (RS) of CORS was 100% for all tests.

Table 1 - Comparison of CORS implementations and direct CRSA on the Dixon-Szegö test functions

Test function	c	$\text{CORS}^{PP(1)}$ (average)	$\text{CORS}^{PP(2)}$ (average)	$\text{CORS}^{PP(3)}$ (average)	CORS [7]	CRSA (average)	CRSA RS
Branin	0.50	19.8	18.4	51.7	34 / 40	138	100 %
Goldstein-Price	0.50	46.5	47.1	59.7	49 / 64	257	100 %
Hartman3	0.50	36.3	33.1	46.7	25 / 61	113	100 %
Shekel5	0.05	57.6	103.4	142.8	41 / 52	621	55 %
Shekel7	0.05	57.3	64.5	94.1	46 / 64	588	55 %
Shekel10	0.05	46.0	62.8	76.6	51 / 64	553	80 %
Hartman6	0.50	102.0	62.6	82.6	104 / 108	427	80 %

The values in the table (except for c and RS) indicate the number of function evaluations to get a relative error less than 1%

The last two columns list the average values and the rate of success RS obtained with the direct application of CRSA.

(1) Results for inverse multiquadric radial basis function

(2) Results for inverse multiquadric radial basis function with a linear polynomial

(3) Results for thin plate spline radial basis function with a linear polynomial (not is necessary an *attenuation factor*)

4. AERODYNAMICS AIRCRAFT WINGS OPTIMIZATION

4.1 Aerodynamics aircraft wings 3D solver

The computational code is based on a first order 3D panel method *(3DPM)* for lift and induced drag coefficient calculation and a 2D boundary layer method *(2DBLM)* for friction coefficient calculation. We selected for the analysis two objectives functions will be the aerodynamic coefficients relations C_L/C_D and the lift coefficient inverse $1/C_L$ as in the solver scheme of figure 1.

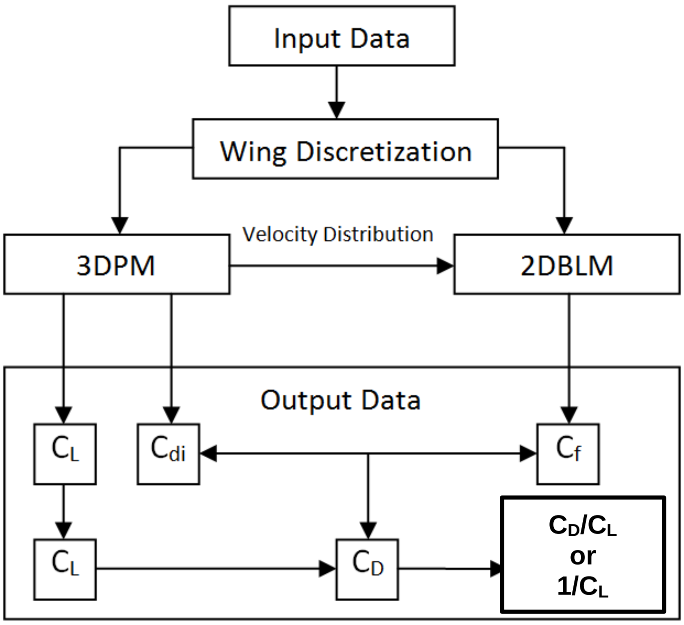

Figure 1. The solvers scheme for *"Black Box Function"*

The *"Solver"* is compound in three sections, the wing and airfoil discretization (i), aerodynamics coefficients calculation (ii) and the objective function calculation (iii). The integration of the three section will be name *"Black Box Function"*.

The wing and airfoil discretization: The airfoil adjustment was performed using two Bezier polynomials of degree 6, one for extrados adjustment and another for intrados. These polynomials are easy to implement in addition to being flexible to adapt to complex geometric shapes and change their shape easily, with the variation of the position of a small number of control points making it suitable for use in optimization problems.

The method uses the Berstein functions, whose equation is shown in Equation (3).

$$B_{i,n} = \binom{n}{i} t^i (1-t)^{n-i} \tag{3}$$

Where "n" represents the Berstein polynomial degree and "t" the parameter contained in the interval [0,1], along the desired curve, then, the airfoil points coordinates can be represented, through the parametric Equations (4):

$$\begin{aligned} x(t) &= \sum_{i=1}^{n} \frac{n!}{i!\,(n-1)!} t^i (1-t)^{n-i} X_i \\ y(t) &= \sum_{i=1}^{n} \frac{n!}{i!\,(n-1)!} t^i (1-t)^{n-i} Y_i \end{aligned} \tag{4}$$

This section was codified in MATLAB 2013a. This code make a resolution of an unrestricted optimization problem that take the original *X* and *Y* airfoil coordinates and then start the iterative *Z* coordinate variation of the five (05) Bezier control point for extrados *(Y_e)* and after another five (05) for intrados *(Y_i)* with the same *X* coordinate fixed lineal array *(X_i)* (as shown in figure 2).

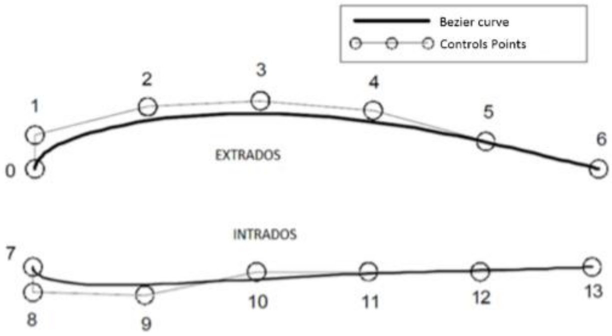

Figure 2: Control point of the Bezier curve.

The iterative process have a double, external and internal loops, the internal is controlled by de *fmincom* Matlab function, and the external, which stop when the distance from the original airfoil coordinates to the Bezier airfoil coordinates (objective function) is minimum.

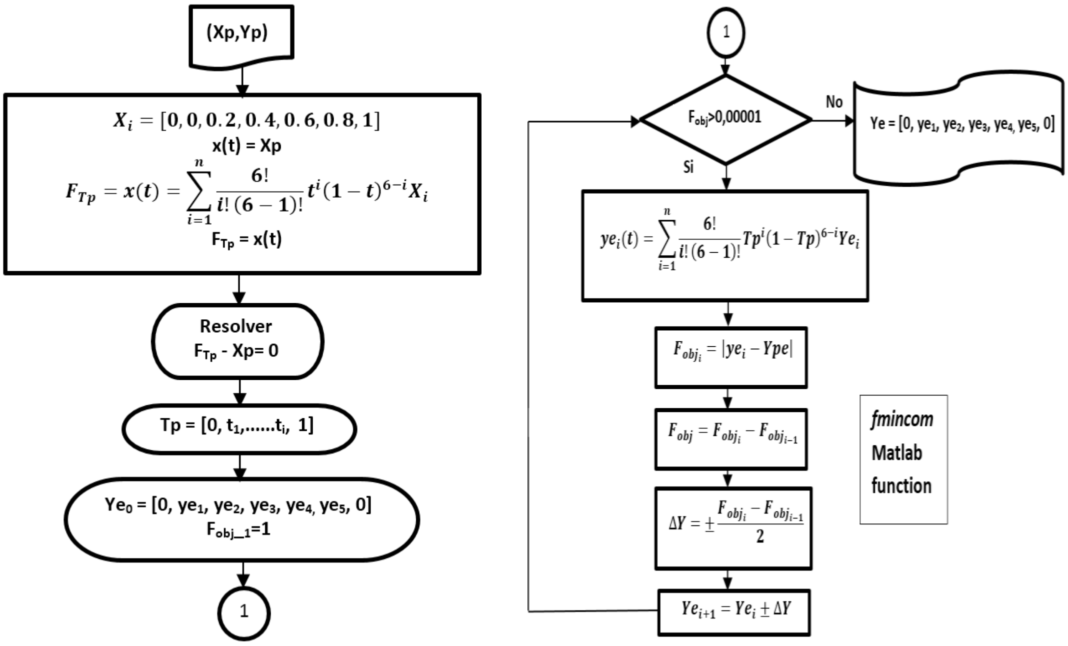

Figure 3: Flow diagram for airfoil adjustment.

The aerodynamic coefficients calculation: The body, immersed in a potential, incompressible and irrotational velocity field, with negligible viscous effects, satisfies the Laplace equation.

$$\nabla^2\Phi^* = 0 \tag{5}$$

where Φ^* is the total potential velocity function, which is the superposition of the potential free-stream velocity function Φ_∞ with the disturbance potential Φ. Then the solution of equation (5) can be constructed by summing a continuous distribution of source singularities σ (thickness effect) and μ dipoles (lifting effect).

$$\Phi^*(x,y,z) = -\frac{1}{4\pi}\int_{s_b}\left[\sigma\left(\frac{1}{r}\right) - \mu\boldsymbol{n}.\nabla\left(\frac{1}{r}\right)\right]dS + \Phi_\infty \tag{6}$$

where S is the body surface. Using the Kutta boundary conditions for the trailing edge and Dirichlet border condition to reduce the flow inside the body to zero. The numerical solution consists in the resolution of the following linear system equation.

$$\sum_{k=1}^{N} C_k\mu_k + \sum_{l=1}^{N_W} C_l\mu_l = -\sum_{k=1}^{N} B_k\sigma_k \tag{7}$$

where C_k and C_l are the influence coefficients for the doublet contributions μ and B_k are the influence coefficient for source contribution σ. Due to the source intensities σ_k are made equal to the normal component of the free-stream velocity.

$$\sigma_k = \boldsymbol{n_k}.\boldsymbol{Q}_\infty \tag{8}$$

where $\boldsymbol{n}$ is the panel normal vector the right hand side of the equation (7) is know, and then the only unknown variables are the doublet intensities. Once known the doublet intensities the velocities at local reference panel can be calculated by differentiation. q_l, q_m and q_n represents the perturbations velocities at the chord wise, span wise and normal direction in each panel

$$q_l = \frac{\partial\mu}{\partial l},\ \ q_m = \frac{\partial\mu}{\partial m}\ \ and\ \ q_n = -\sigma, \tag{9}$$

Then the total velocity distribution can be expressed by the equation (10)

$$\boldsymbol{Q}_k = (Q_{\infty l}, Q_{\infty m}, Q_{\infty n})_{\boldsymbol{k}} + (q_l, q_m, q_n)_{\boldsymbol{k}} \tag{10}$$

Once the velocities have been obtained, the pressure coefficient distribution Cp_k is calculated immediately by the equation (11).

$$Cp_k = 1 - \frac{Q_k^2}{Q_\infty^2} \tag{11}$$

Finally the aerodynamics force is calculate by the equation (12) where ρ is the air density and ΔS_k is the panel area.

$$\Delta\boldsymbol{F}_k = Cp_k\left(\frac{1}{2}\rho Q_\infty^2\right)\Delta S_k\boldsymbol{n}_k \tag{12}$$

The objective function calculation: The lift (C_L), drag (C_D), and moment (C_M), coefficients of the wing are calculated using a code adapted from Katz & Plotkin (CKP) based on the first order 3D panels method as shown in "Solver scheme" figure 1.

The adaptation was realized in three fundamental aspects that are:

a. The matrix sizing of the variables with the purpose of being able to handle discretized wings of up to 10000 panels.
b. Creation of an output data file with velocity distribution to link to the boundary layer code, which calculates the friction drag coefficient *(Cf)* on the wing, this code is based on the boundary layer theory.
c. Creation of an output data file with the values of the functions S_1 and S_2 attending in section 4.3, which will be taken by the optimizer as *objective function.*

4.2 Aerodynamics wing solver test.

In this section will be shown the results obtained by the aerodynamics solver program obtained in a wing model with the following geometric characteristics as shown in the figure 4 from Chao, L (2007).

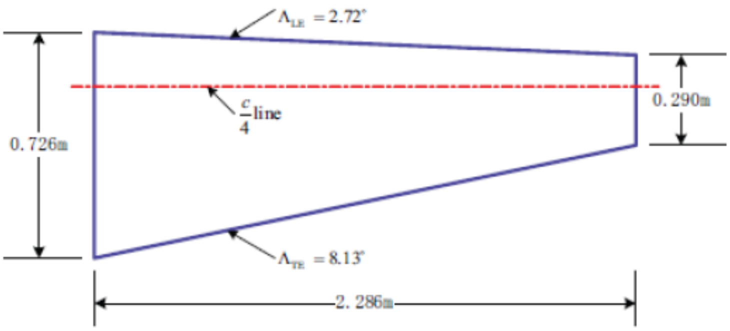

Figure 4. Wing geometry with airfoil NACA 65-210 for test (Chao, L (2007)).

Table 2 - Results of Lift coefficient C_L, Drag Coefficient C_D and Aerodynamics efficiecy η

α (°)	C_L	C_D	η	α (°)	C_L	C_D	η
0	0,14	0,0155	9,0323	9	0,82	0,0314	26,1146
1	0,215	0,0161	13,3540	10	0,93	0,0339	27,4336
2	0,318	0,0169	18,8166	11	1,05	0,0394	26,6497
3	0,4	0,0177	22,5989	12	1,08	0,0415	26,0241
4	0,44	0,0188	23,4043	13	1,15	0,0485	23,7113
5	0,51	0,0208	24,5192	14	1,2	0,0545	22,0183
6	0,62	0,0231	26,8398	15	1,28	0,0671	19,0760
7	0,7	0,0244	28,6885	16	1,3	0,0801	16,2297
8	0,79	0,0277	28,5199				

The results were verified by experimental data used by Chao, L (2007). The comparative curve of C_L vs α and C_D vs α are showing in the figure 5a and 5b.

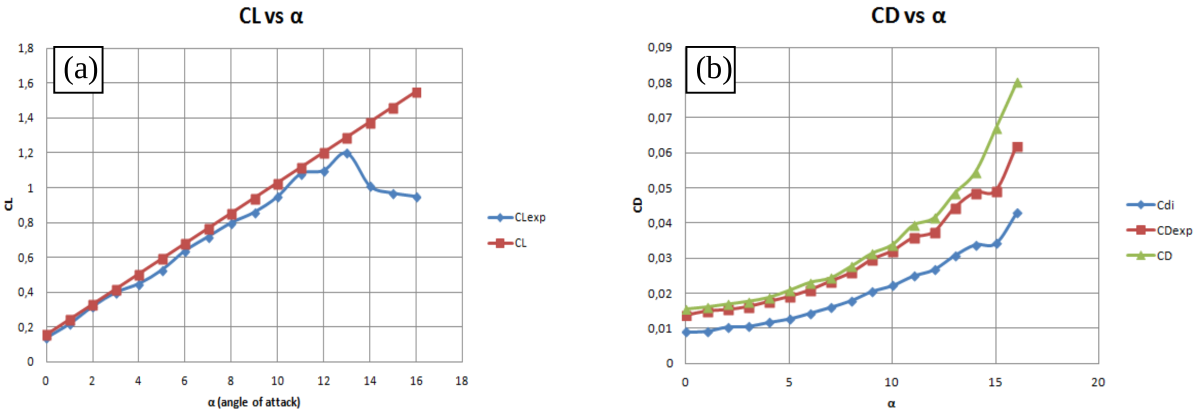

Figure 5: Comparative curve of C_L (a); Comparative curve of C_D (b)

4.3 Aircraft wing optimization problem.

It is important to establish the behavior of the CRSA and CORS algorithm on aerodynamics wing design problems. It was performs first, the wing parameters optimization, based on CRSA and after with CORS, both case for two interesting objective functions (S_1 and S_2).

Aerodynamic optimization problems definitions:

In this optimization problem, the variable space $\vec{X}$ is conformed by the geometrics wing parameters, such as the root chord (C_o), the tip chord (C_t), the sweep angle (φ), the dihedral angle (ε) and the wingspan (b). The objectives functions to be minimized are the inverse of the lift coefficient ($S_1=1/C_L$) and the aerodynamics coefficients relation ($S_2=C_D/C_L$). For both case were used 3600 panels, divided in 200 points on the airfoil shape line ($Np=200$) and 18 wing span divisions ($JB=18$) divisions.

These parameters were input by user data file, like *the wing base parameters* (C_o=1 m, C_t=1 m, φ=0°, ε=0° and b=10 m.). Based on this wing was obtained the solutions for lowers and uppers bounds of lateral constrains. It was exploring, increments between 80% and 200% on root chord of the base wing value, taper ratio (λ) between 0 and 1, minimum sweep angle of 0 ° and maximum of 45 °, minimum dihedral angle of 0 ° and maximum of 5 ° and increment between 50% and 150% on wing span of the base wing value.

The optimization problem for this case is mathematically defined as follows:

With $\vec{X} = X(C_0, C_t, \varphi, \varepsilon, b)$

minimize $S_1(\vec{X}) = \dfrac{1}{C_L}$ or $S_2(\vec{X}) = \dfrac{C_D}{C_L}$

Suject to the following lateral constrains R_1 to R_5:

R_1: $0{,}8m \leq C_0 \leq 2$ m

R_2: $0 \leq \lambda \leq 1$

R_3: $0° \leq \varphi \leq 45°$

R_4: $0° \leq \varepsilon \leq 5°$

R_5: $5\,m \leq b \leq 15$ m

4.4 Wings parameters optimization problems with CRSA.

The optimizer used by Silva, 2009, codified in FORTRAN, was linked with the solver "Dublet_3DD" showed in figure 1, following the next scheme in figure 6.

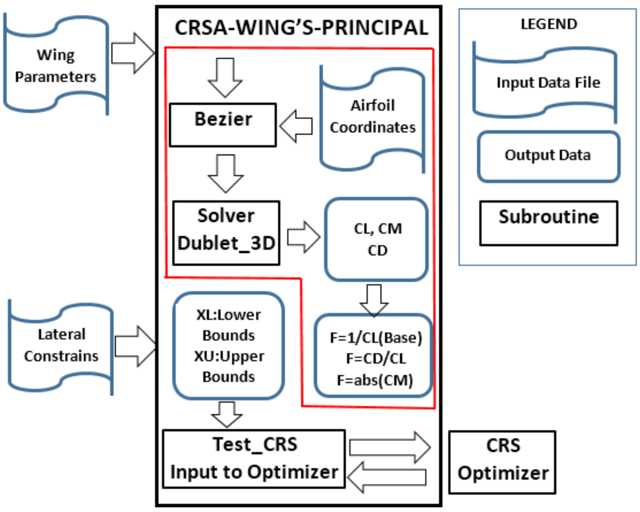

Figure 6: CRSA-Doublet 3D solver interface diagram

The CRSA code tested was run 10 times in order to obtain the best result, taking in account that the CRSA is a population stochastic algorithm. The figure 7, show the best value in the objective function evaluation and the objective function calling for each run and the average for both objective functions.

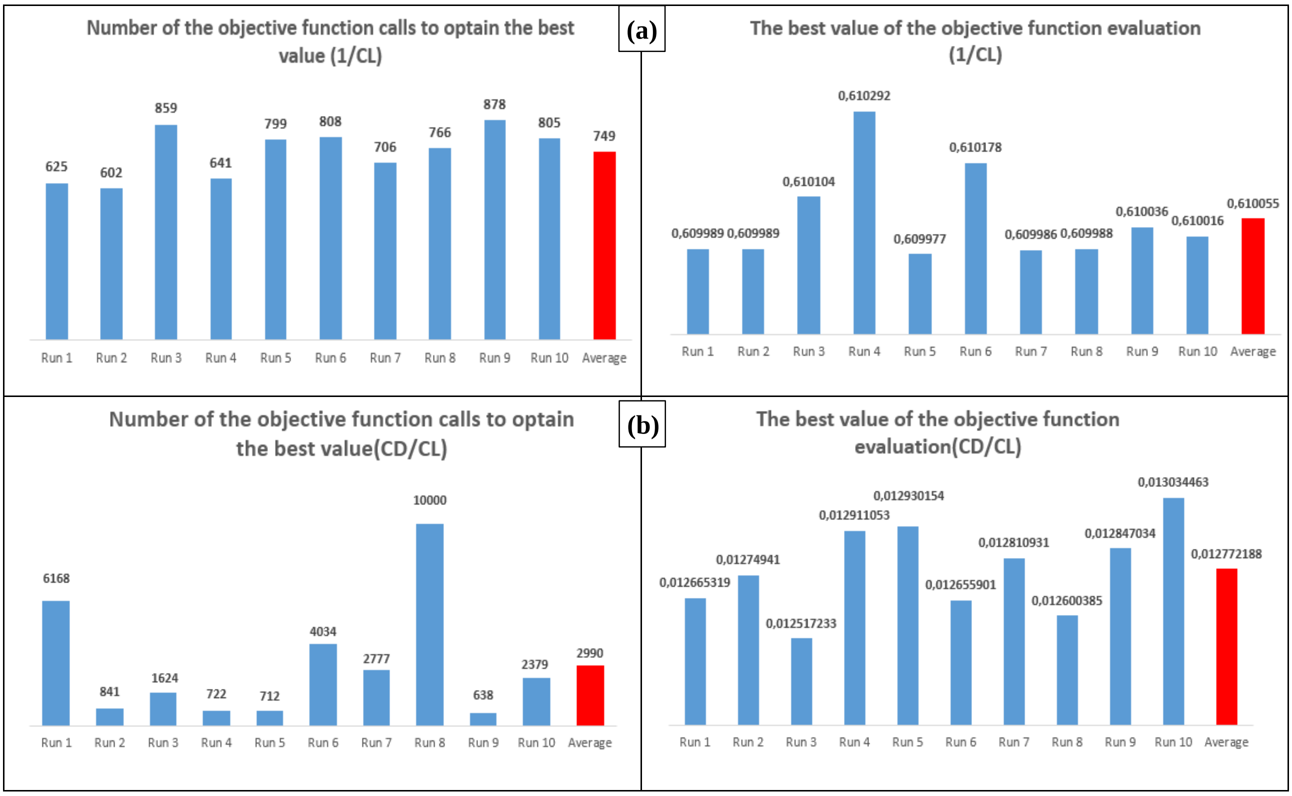

Figure 7. The minimum and the objective function calling for each CRSA run (a) $1/C_L$ (b) C_D/C_L.

Based on the figure 9a, the "Run 2" was taken as the best point result of CRSA ($CRSA^{BR}$), with a minimum objective function value of 0,609989 in 602 calls of solver. Using the same figure, for the objective function ($1/C_L$); The average percentage value of the differences between each run and the average of all is around 11,26% for calls and 0,013% for S_1

value. In this case the optimum for CRSA is $\vec{X} = (1.81511 \;\; 0{,}30009 \;\; 8{,}70311 \;\; 1{,}99207 \;\; 14{,}99989)$ and $S_1(\vec{X}) =$ 0,609989 is obtained in 602 objective function calls.

For the objective function *(C_D/C_L)* the average for minimum converged of the objective function value is 0,0127721 in 2990 objective function calling. Based on the figure 9b, the "Run 3" was taken as the best run result of CRSA; The average percentage value of the differences between each run and the average of all is around 75,15% for calls and 1,05% for S_2 value. In this case, the optimum for CRSA is $\vec{X} = (0{,}80067 \;\; 0{,}300000 \;\; 44{,}41187 \;\; 1{,}40151 \;\; 14{,}98552)$ and $S_2(\vec{X}) = 0{,}01252$ is obtained in 1624 objective function calls.

The figure 8a and 8b shown the convergence graph for CRSA ten runs and the best run for S_1 and S_2 objective functions. The convergence of the objective function value $S_1 \approx 0{,}61$ is obtained around 600 calling with a time/calling of 5,60 seg/call; while for the objective function $S_2 \approx 0{,}0126$ is obtained around 2500 calling with a time/calling of 5,62 seg/call.

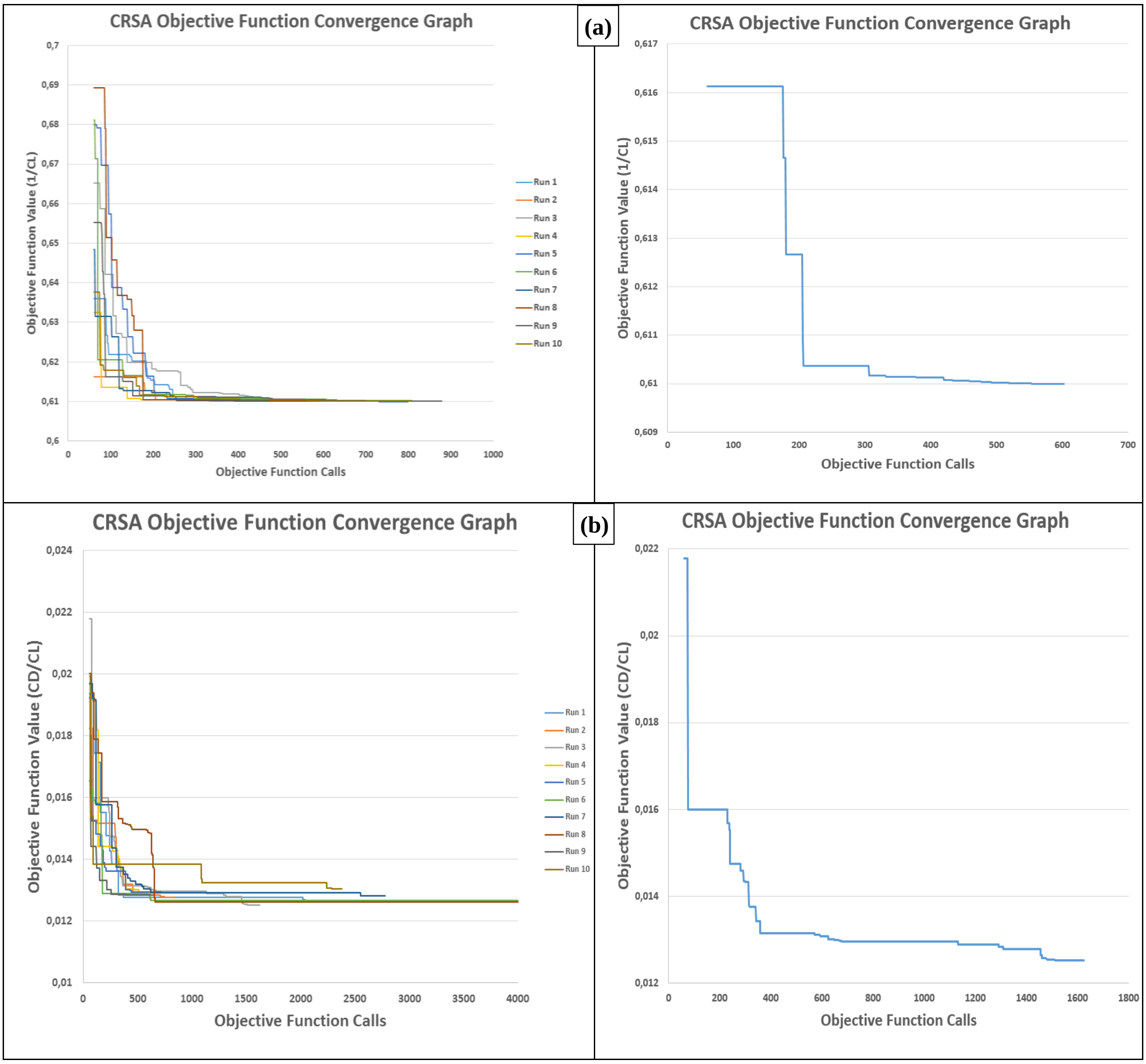

Figure 8. The minimum and the objective function calling for each CRSA run (a) $1/C_L$ (b) C_D/C_L.

4.5. Wings parameters optimization problem with CRSA+CORS

The CORS implementation on the wings parameters optimization problem was using the same parameters used for Dixon-Szegö test functions. The CRSA was applied 10 times, for metamodel optimization with a population equal to 60 is adopted. The computational experimental procedure we propose here is described by Silva (2009). The radial basis function used is the inverse multiquadric with the search pattern ⟨0.95,0.50,0.25,0.005,0.0005,0.00⟩, N=6. The comparative graphics between the best run CRSA and CORS application for the objective functions $S_1=1/C_L$ and $S_2=C_D/C_L$ are shown in figure 9a and 9b. For CORS implementation were taken into account two cases, in the *case 1*, CORS would remain running until to attend 350 iterations in order to observe the number of the calls to reach the convergence, in order to do a comparison with the convergence values using CRSA. With both objectives functions, CORS manages to achieve a better optimal point in less calls of the objective function, as shown in figure 10a and 10b for S_1 and S_2 objectives functions.

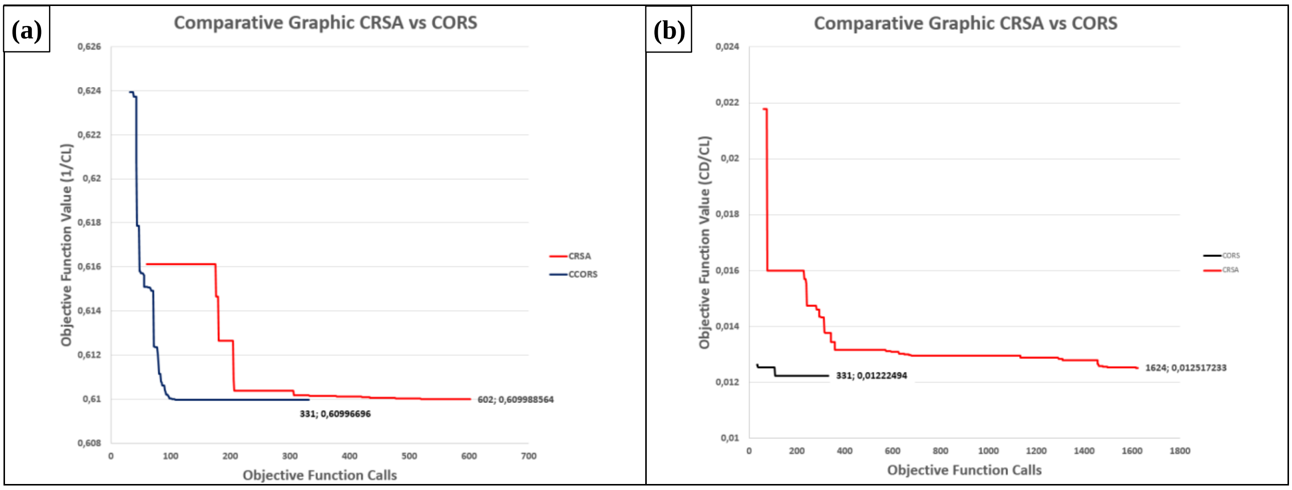

Figure 9. The CRSA and CRSA+CORS comparative graphic using objectives functions (a) $1/C_L$ (b) C_D/C_L.

The optimum for CORS is $\vec{X} = (1{,}75771 \;\; 0{,}30023 \;\; 9{,}22613 \;\; 0{,}00000 \;\; 14{,}99989\,)$ and $S_1(\vec{X}) = 0{,}60997$ obtained in 331 objective function *(S₁)* calls, and for the objective function *(S₂)*, The optimum for CORS is $\vec{X} = (0{,}80018 \;\; 0{,}300000 \;\; 44{,}41187 \;\; 0{,}00000 \;\; 15{,}00000)$ and $S_1(\vec{X}) = 0{,}012224935$ obtained in 331 calls.

In *case 2*, CORS would remain running until to attend at less 100,01% of the CRSA best run value, with the purposed of measuring the number of the calls needed for CORS to reach it, in order to calculated the acceleration provided by the use of CORS ($CORS^{ACC}$). The procedure to obtain this is describe below.

Step 1: calculate the 100,01% of the CRSA objective function value, to be reach ($CRSA^{100{,}01\%}$)

$$CRSA^{100{,}01\%} = 1{,}0001 \; x \; CRSA^{BR} \tag{13}$$

Step 2: explore the CRSA results from the figures 10a and 10b to obtain the numbers calls necessary to reach the $CRSA^{100{,}01\%}$. ($NC\text{-}CRSA^{100{,}01\%}$)
Step 3: run the CORS until to attend the $CRSA^{100{,}01\%}$ as a stop criteria, to obtain the numbers of calls necessary to do it ($NC\text{-}CORS^{100{,}01\%}$)
Step 4: calculate the acceleration improve by CORS.

$$CORS^{ACC} = \frac{NC - CRSA^{100{,}01\%}}{NC - CORS^{100{,}01\%}} \tag{14}$$

The results of the case 1 and case 2 are show in the table 3

Table 3 - Comparison of CORS implementations and direct CRSA on the wing parameters problem

Objective function	*c*	*$CRSA^{BR}$ (best run)*	*CORS*	*$NC\text{-}CRSA^{100{,}01\%}$*	*$NC\text{-}CORS^{100{,}01\%}$*
$S_1=1/C_L$	0.50	602 / 0,60999	331 / 0,60997	459	98
$S_2=C_D/C_L$	0.50	1624 / 0,01252	331 / 0,01222	1616	108

Using the equation (14) the acceleration improve by CORS for the objectives functions S_1 and S_2 are 4,68 and 14,96 respectively.

5. CONCLUDING REMARKS

The CORS strategy presented in this paper was applied on the Dixon-Szegö test functions. The results have shown that the strategy is promising in accelerating and improving the reliability of CRSA. The CORS was able not only to accelerate the optimization of costly functions but also to improve the reliability and robustness of the search process.

The tested 3D aerodynamic solver is relatively fast and produces values of C_L and C_D in acceptable agreement with experimental results for low angles of attack. It will be useful for testing CORS methodology with the ultimate goal of using it with higher fidelity solvers based on CFD.

The objective function $S_1=1/C_L$ is well behaved to the CRSA application in aerodynamics optimization problem that only involved wing parameters, whereas, the objective function $S_2=C_D/C_L$ takes a long time to reach the convergence, therefore, this function is sensitive to changes, due to small variations on wing parameters. The CORS application in this function improves notoriously the convergence rate, obtaining better values of the objective function with much lower numbers of objective function calls.

With the CORS application, the optimization process is accelerated about 4,7 times more than CRSA for objective function S_1 and 14,96 times for the objective function S_2, concluding that aerodynamics problems can be accelerated with this methodology.

The use of only wing parameters as solution space provides solutions close to or just on the upper and lower limits of the feasible region. The inclusion of the airfoil control points from Bezier adjustment should be improve optimal solutions not just on the borders of the feasible region.

In this work, the CORS methodology was applied in aerodynamic wing optimization problem using a low fidelity solver based on 3D Panel Method. Satisfactory results were obtained in terms of problem solution acceleration and attainment of global optimum. These results indicate a promising perspective for the use of the CORS strategy with a high fidelity codes based on CFD in the resolution of aerodynamics optimization problems.

6. ACKNOWLEDGEMENTS

The authors would like to thank UNIFEI (Universidade Federal de Itajubá) for the logistical support in the use of its facilities and equipment in the development of this research and especially to FAPEMIG (Fundação de Amparo à Pesquisa do Estado de Minas Gerais), for the economic and financial support received.

7. REFERENCES

Ali, M. M., Törn, A. and Viitanen, S., 1997, "A Numerical Comparison of Some Modified Controlled Random Search Algorithms", Journal of Global Optimization, Vol. 11, 377-385.

Chao, L. 2007. "Wake Vortex Encounter Analysis with Different Wake Vortex Models Using Vortex-Lattice Method".Delft University of Technology.

Dixon, L.C.W. and Szegö, G., 1978, "The global optimization problem: an introduction, Towards Global Optimization", L.C.W. Dixon, G. Szegö (Eds.), Vol. 2, 1–15. North-Holland, Amsterdam.

Katz and Plotkin, 1991, "Low Speed Aerodynamics", Ed. McGraw-Hill, (ISBN 0-07-100876-4).

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, J. Herskovits, S. Mazorche, A. Canelas (Eds.), ISSMO, Rio de Janeiro, Brazil, 2005 (ISBN: 85–285–0070–5).

Moran. J., 1984, "An introduction to theoretical and computational aerodynamics", Ed. Jhon Wiley & Sons, (ISBN: 0-471-87491-4).

Praveen, C. and Duvigneau, R., 2007, "Radial Basis Functions and Kriging Metamodels for Aerodynamic Optimization", RR-6161, 40 p., INRIA, France.

Regis, R.G. and Shoemaker, C.A., 2005, "Constrained global optimization using radial basis functions", Journal of Global Optimization, Vol. 31, 153–171.

Silva, E. Manzanarez-Filho, N. Ramirez, R and Lima, A., 2009, "Metamodeling approach using radial basis funtions and a controlled random search with application on the controlled cascade design", 20th International Congress of Mechanical Engineering (COBEM 2009).

8. RESPONSIBILITY NOTICE

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