31st Congress of the International Council
of the Aeronautical Sciences

Belo Horizonte, Brazil; September 09-14, 2018

COMPARISON AND APPLICATION OF DIFFERENTS TECHNIQUES OF MULTI-OBJECTIVE OPTIMIZATION FOR AERODYNAMIC WING DESIGN

Nelson Jose Diaz Gautier[1,2], Nelson Manzanares-Filho[1], Edna Raimunda Da Silva[1]
[1]Federal University of Itajubá - Brazil UNIFEI, [2]National University of the Army Force – Venezuela UNEFA

Abstract

In the real world, aeronautics engineering design problems the designer often seeks to optimize multiple and conflicting merit functions or objectives relating to the performance of the given design. Several techniques are available today for design through numerical optimization. The first aim of this paper was to perform a complete, comprehensive comparison between different stochastic multi-objective optimization methods like: the Non-dominated Sorting Genetic Algorithm II (NSGA-II). A single and Multi Objective Simulated Annealing (MOSA). Multi-Objective Particle Swarm (MOPSO) and Multi Objective Genetic Algorithm (MOGA-II), by using the commercial Mode FRONTIER software to solve the aerodynamics wing design problem. The black-box objective function is a low-fidelity flow computation code solver, which was based on first order 3D panel method, for lift and induced drag coefficient calculation, and the boundary layer method for the friction drag coefficient calculation. The efficiency of each method was determined, taking in account three numerical performance metrics and one visual criterion for qualitative and quantitative comparison. (1) The ratio between the number of resulting Pareto front members and the total number of fitness function calculations, (2) Spacing metric S which indicates the distribution of the Pareto front in the objective space, (3) the generational distance for the convergence and finally the graphical representation of the Pareto fronts for discussion. These metrics were chosen to represent the quality, as well as the speed of the algorithms by ensuring well distributed solution.

1 Introduction.

The mono and multi-objective optimization problems applied to the wing aerodynamic design involve a large number of variables. The influence of multi-objective optimization on aerodynamic design is a field of research to be investigated. It should be taken into account that such optimization methodologies involve different procedures to evaluate a set of aerodynamic design variables. In optimization problems, these procedures can be called "black box functions", and are often addressed in the literature as design methods, IE, low-fidelity methods such as the Prandtl-line method, half-fidelity methods such as basic 3D or viscous panel methods and High-fidelity Computational Fluid Dynamics (CFD).

The present study investigates the behavior of the different Multi-Objective Optimization Algorithms (MOOA) applied in the aerodynamic design. Generally, a method cannot be declared better than others without applying to a specific problem that needs to be confronted. The important features that need to be evaluated include: 1) generality of formulation versus problem dependence; 2) robustness, intended as the ability to avoid local optimum, versus the need for interaction and human knowledge; 3) multiple goal optimization capability versus single goal; and 4) computational efficiency versus the need for large computational resources. In order to determine the previous characteristics, the NSGA II, MOSA, MOPSO and MOGA II multi-objective optimization algorithms will be applied to the wing aerodynamic problem, a process integration commercial software will be used to meet this objective.

2 Aerodynamics Solver

We are going to use a computational methodology code developed by [1]; it was based on first order 3D panel method *(3DPM)* for lift and induced drag coefficient calculation and 2D boundary layer method *(2DBLM)* for friction coefficient calculation and then the aerodynamic efficiency as in the solver scheme of the Figure 1.

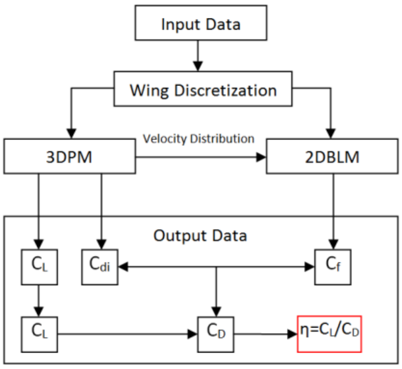


Fig.1. Solver Scheme.

2.1 The input and output data of the Solver

The input and output Data includes in the algorithm like a text or data file, it´s in order to receive or send data in iterative form to the optimization process. The inlet data are listed in the Table 2, from this table will be selected the optimization problem variables or design variables.

Table 2 - The Input and Output Data set by the user

Input data		Output data	
Root Chord	C_0	Lift	C_L
Tip Chord	C_t	Induced Drag	C_{di}
Wing Span	b	Friction Drag	C_f
Airfoil NACA Digit	XXXX	Total Drag	C_D
Panels on airfoil	IB	C_D/C_L	η
Wing Span Division	JB1		
X Root Coordinate	XR		
X Tip Coordinate	XT		

2.2 Preprocessing part or wing discretization

Previously to the wing discretization, the airfoil is divided in IB panels using a MATLAB code based on Bezier curves to generate the airfoil shape and discretization as shown in Figure 3.

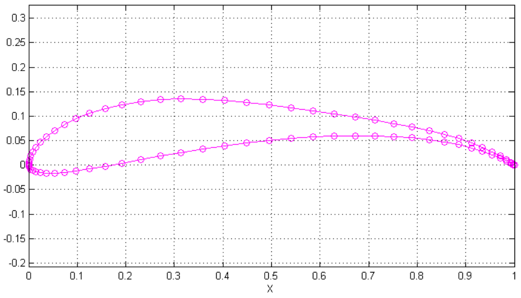


Fig.2. Airfoil S1223 panels discretization

Extending the airfoil discretization along the wingspan was obtained the wing panels discretization as shown in Figure 3. It was using the 3D Panel Method codified in FORTRAN.

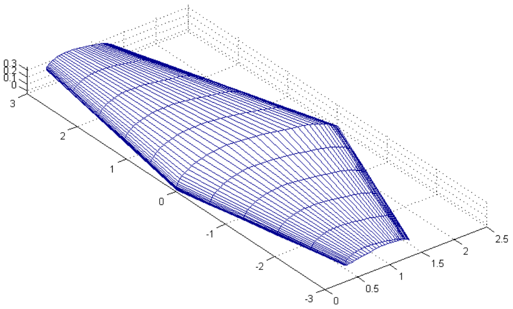


Fig.3. Wing discretization

3 Multi-objective optimization algorithm´s

Whilst most real world problems require the simultaneous optimization of multiple, often competing criteria (or objectives), the solution to such problems is usually computed by combining then into a single criterion to be optimized, according to some utility function. In many cases, however, the utility function is not well known prior to the optimization process the whole problem should then be treated as a multi-objective problem with non-conflicting objectives. In this way, a number of solutions can be found which provide the decision maker (DM) with insight into the characteristics of the problem before a final solution is chosen.

Multi-objective optimization (MOO) seeks to optimize the components of a vector value cost function. Unlike the single objective optimization, the solution to this problem is not a single point, but a family of points known as the Pareto optimal set.

3.1 Non dominated Sorting Genetic Algorithm (NSGA II)

The non-dominated sorting GA (NSGA) proposed by [2] has been applied to various problems. However, there have been a number of criticisms of the NSGA. The NSGA II, modified the first NSGA approach in order to alleviate all the difficulties. The NSGA-II is formed by the follow modules.

3.1.1 A fast non-dominated sorting approach.

In order to sort a population of size N according to the level of non-domination, each solution must be compared with every other solution in the population to find if it is dominated.

3.1.2 Density estimation.

To get an estimate of the density of solutions surrounding a particular point in the population, we take the average distance of the two points on either side of this point along each of the objectives. This quantity serves as an estimate of the size of the largest cuboid enclosing the point /without including any other point in the population (*crowding distance*). See figure 3

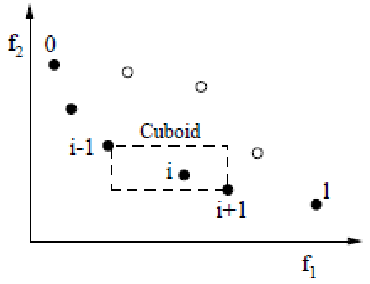


Fig.4. Crowding distance

3.1.3 Crowded comparison operator

The crowded comparison operator guides the selection process at the various stages of the algorithm towards a uniformly spread out Pareto optimal front. Let us assume that every individual *I* in the population has two attributes: *Non-domination rank* and *local crowding distance.*

3.1.4 The main loop

Initially, a random parent population P_0 is created. The population is sorted based on the non-domination. Each solution is assigned a fitness equal to its non-domination level (1 is the best level). Thus, minimization of fitness is assumed. Binary tournament selection, recombination, and mutation operators are used to create a child population Q_0 of size *N*. From the first generation onward, the procedure is different.

3.2 Multi Objective Simulated Annealing (MOSA)

The basic structure of the MOSA algorithm developed by [3] its novel features over that of traditional single objective optimization by Simulated Annealing are described as follows.

A "composite" objective function G is formed between each of the single objectives and is defined as

$$G = \sum_{i=1}^{M} \ln f_i \qquad (1)$$

where $f_1, \dots, f_M$ are the objective functions to be minimized. At any given iteration, *n*, the objective function values are calculated from control variable values stored in X_n, and are substituted into Equation (2) to give $G(X_n)$. Random perturbation is then applied to each variable, from which $G\ (X_{n+1})$ is computed, until an acceptance criterion is attained. The acceptance criterion with the following probability, p, is employed between successive iterations.

$$p = exp\left(-\frac{1}{T}[G(X_{n+1}) - G(X_n)]\right) \qquad (2)$$

where T, as before, is the system temperature. The expression in Eq. (2) Is such that any perturbation which decreases G enables X_{n+1} as the current position from which to initiate a further search, while perturbations which increase G are accepted with a probability which falls as the temperature is lowered as the search progresses. Using Eq. (1), Eq. (2) can be written as:

$$p = exp\left(-\frac{1}{T}\sum_{i=1}^{M} ln\left[\frac{f_i(X_{n+1})}{f_i(X_n)}\right]\right) \qquad (3)$$

During the course of the search, an "archive" of non-dominated solutions $f_1, \dots, f_M$ is also maintained.

3.3 Multi-Objective Particle Swarm Optimization (MOPSO).

The important part in multi-objective particle swarm optimization (MOPSO) is to determine the best global particle for each particle of the population. In the mono objective particle, swarm optimization method (PSO) the global best particle is easily determined by selecting the particle, which has the best position. Since in multi-objective optimization problems there are a set of Pareto optimal solutions as the optimum solutions, each particle of the population should select one of the Pareto-optima's as its global best particle, which we call it the best local guide.

PSO is the expansion of animal social behavior that follows a population-based meta-heuristic strategy for optimization. It incorporates the acceleration by distance and velocity matching by nearest matching. In the mid- 1990s, it was originally introduced by [3], it was employed to balance the weights in neural networks [4].

In his work, Kumar [5] present a glossary of common terminology and their definitions follows, for clarity:

a). *Swarm:* Population size (number of particles).

b). *Particle:* An individual member of the swarm, which is a potential solution to the problem.

c). *pbest (personal best):* The personal best position achieved by a particle so far.

d). *lbest (local best):* Position of the best particle member within the neighborhood.

e). *gbest (global best):* Position of the best particle member from the entire swarm.

f). *Leader:* A particle that guides the other particles of the swarm to the best regions in the search space.

g). *Velocity (vector):* The direction in which a particle must move in order to improve its position.

h). *Inertia Weight:* The impact of the previous velocity to the current velocity of the particle, denoted by w.

i). *Learning Factor:* The attraction of a particle towards either its previous or its neighbors' values. PSO adopts two learning factors, *cognitive learning factor* ($c1$) and *social learning factor* ($c2$). Cognitive leaning factor represents the attraction toward the particle's own success, and social learning factor represents the attraction toward neighbors. Both factors are constants usually considered in the experiments.

j). *Neighborhood Topology:* The set of particles involved to determine the *lbest* value of the given particles.

The position and velocity of the particle at each iteration *I* can be calculated by the equations (4) and (5)

$$X_i(t+1) = X_i(t) + V_i(t+1) \qquad (4)$$

$$V_i(t+1) = V_a + V_l + V_g \qquad (5)$$

where, V_a, V_l and V_g are the velocities for search the best personal, local and global positions: $V_a = V_i(t)$, the actual swarm velocity, $V_l = c_1 r_{i,1}(t) x \left(pbest_{i,t}(t) - X_i(t)\right)$, the local and $V_g = c_2 r_{i,2}(t) x \left(gbest_{i,t}(t) - X_i(t)\right)$ the global.

The algorithm and pseudo code of MOPSO are detailed in [5].

3.4 Multi-Objective Genetic Algorithm (MOGA-II).

This is an improved version of MOGA (Multi-Objective Genetic Algorithm) by [6] and is should not be confused with MOGA by [7] with which it shares only the same acronym. MOGA II is a smart multi-search elitism for robustness and directional crossover, for fast convergence. Its efficiency in ruled by its operators (classical crossover, directional crossover, mutation and selection) and by the use of elitism.

Encoding in MOGA-II is done as in classical genetic algorithms and it uses four different operators for reproduction: classical crossover, directional crossover, mutation and selection. At each step of the reproduction process, one of the four operators is chosen (with regard to the predefined operator probabilities) and applied to the current individual. Directional crossover assumes that a direction of improvement can be detected comparing the fitness values of two reference individuals. In [8] a novel operator called evolutionary direction crossover was introduced and it was shown that even in the case of a complex multimodal function this operator outperforms classical crossover. The direction of improvement is evaluated by comparing the fitness of the individual Indi from generation t with the fitness of its parents belonging to generation t − 1. The new individual is, then created by moving in a randomly weighted direction that lies within the ones individuated by the given individual and his parents. A similar concept can be however applied The MOGA-II algorithm is shown at the reference [9].

4 Multi-objective Aerodynamics wing optimization problem

In this paper, two multi-objective aerodynamic optimization problems are characterized considered by the authors that involve a great number of the wing geometric variables and the more important aerodynamics coefficients.

4.1 Only geometric wing parameter variation (Problem 1).

In order to have in the methodology a representative number of geometric variables that can define the shape of the wing, the following decision variables are selected:

- Wingspan (b)
- Wing root chord (C_0)
- Wing tip chord (C_t)
- Sweep angle (φ)
- Dihedral angle (ε)

The airfoil wing coordinates remain invariables in this problem. Then the n-dimensional vector that represents each individual will be:

$$\vec{X}^5 = X(b, C_0, C_t, \varphi, \varepsilon) \tag{6}$$

In this problem two objective functions will be analyzed: the inverse of lift coefficient (*f₁*) and longitudinal moment coefficient (*f₂*)

$$f_1 = \frac{1}{C_L} \quad o \quad f_2 = C_m \tag{7}$$

The constrains equations R_n will be lateral constrains type: the first one is related to the wing aspect ratio (Λ):

$$\Lambda_{min} \leq \frac{b^2}{S_w} \leq \Lambda_{max} \tag{8}$$

where S_w is the wing surface; $S_w = \frac{b}{2}(C_0 + C_t)$

$$R_1\!: \ \Lambda_{min} \leq \frac{2b}{(C_0 + C_t)} \leq \Lambda_{max} \tag{9}$$

The second constraint is the taper ratio λ, which must be less than unity,

$$R_2\!: \ \frac{C_t}{C_0} \leq 1 \tag{10}$$

The third constraints are the sweep angle (φ),

$$R_3\!: \ \varphi_{min} \leq \varphi \leq \varphi_{max} \tag{11}$$

The fourth and last constrain is in the dihedral angle.

$$R_4\!: \ \varepsilon_{min} \leq \varepsilon \leq \varepsilon_{max} \tag{12}$$

4.2 Only airfoil wing shape parameter variation (Problem 2).

Diaz *et al* in [1], designed a code for airfoil shape adjustment, based on Bezier curve. The components of the Z_{PC} vector represent the abscissas of the control points to generate the wing profile.

$$\overrightarrow{Z_{PC}} = (Z_1, Z_2, Z_3, Z_4, Z_5, Z_8, Z_9, Z_{10}, Z_{11}, Z_{12}) \tag{13}$$

In this problem the same objectives functions of the problem 1 are used. The lateral constrains was taken in order to maintain the leading edge rounded and the max chamber to avoid the boundary layer separation [10].

$$\begin{aligned}
&R_1: Z_{1B} - 0{,}1|Z_{1B}| \leq Z_1 \leq Z_{1B} + 0{,}1|Z_{1B}| \\
&R_2: Z_{2B} - 0{,}5|Z_{2B}| \leq Z_2 \leq Z_2 + |Z_{2B}| \\
&R_3: Z_{3B} - 0{,}5|Z_{3B}| \leq Z_3 \leq Z_{3B} + |Z_{3B}| \\
&R_4: Z_{4B} - 0{,}5|Z_{4B}| \leq Z_4 \leq Z_{4B} + |Z_{4B}| \\
&R_5: Z_{5B} - 0{,}1|Z_{5B}| \leq Z_5 \leq Z_{5B} + 0{,}1|Z_{5B}| \\
&R_6: Z_{8B} - 0{,}1|Z_{8B}| \leq Z_8 \leq Z_{8B} + 0{,}1|Z_{8B}| \\
&R_7: Z_{9B} - 0{,}5|Z_{9B}| \leq Z_9 \leq Z_{9B} - 0{,}25|Z_9 B| \\
&R_8: Z_{10B} - 0{,}5|Z_{10B}| \leq Z_{10} \leq Z_{10B} - 0{,}25|Z_{10B}| \\
&R_9: Z_{11B} - 0{,}5|Z_{11B}| \leq Z_{11} \leq Z_{11B} - 0{,}25|Z_{11B}| \\
&R_{10}: Z_{12B} - 0{,}1|Z_{12B}| \leq Z_{12} \leq Z_{12B} + 0{,}1|Z_{12B}|
\end{aligned} \tag{14}$$

5 Performance Metrics

In Multi-Objective Optimization Processes (MOPs), there are two distinct and orthogonal goals as follows: (1) to discover solutions as close to the Pareto-optimal solutions as possible, and (2) to find solutions as diverse as possible in the non-dominated front obtained. Therefore, in order to compare two or more MOEAs, at least two performance metrics (one evaluating the progress towards the desired Pareto-optimal front and the other evaluating the spread of Pareto-front obtained) need to be used and exact definitions of these performance metrics are important.

5.1 Performance Metrics Classification for Multi-Objective Optimization (MOO).

This section covers the current performance metrics in the literature of EMOO (Elitistic Multi-Objective Optimization). The main purpose of a performance indicator in EMOO is to quantify the performance from a specific point of view. The ultimate goal in EMOO is to find a very accurate approximation and large number of the true Pareto optimal solutions with uniform distribution among all objectives [11]. Therefore, classify the current performance measure scan into three main categories: capacity, coverage and convergence metrics.

5.1.1 Capacity performance metrics:

This group of metrics tally the number or ratio of non-dominated solutions in the objective function space that satisfies given predefined requirements. In addition, the number of Pareto optimal solutions obtained is also important [12], which provides decision making with more designs from which to choose. The capacity metric selected to use in this paper is:

Perceptual Hit Rate Metric (HRM%): the number of resulting Pareto front point is given by $|P^*|$, while the parameters *FC* denotes the total number of fitness calculation. The final hit rate HRM% is computed according to the equation (15)

$$HRM\% = \frac{|P^*|}{FC} 100[\%] \tag{15}$$

A higher *HRM*% indicates that fewer time-consuming fitness computations were used to find the Pareto optimal solutions. The relationship between the size of the feasible design space and the ideal Pareto front should be considered in order to create a universal success quantifier.

5.1.2 Coverage performance metrics:

This class of metrics defines how well the solutions obtained "cover" the range of each of the objectives. The coverage metric selected to use in this work is:

Spacing Metric (SM): This metric as used by [13], indicates how uniformly the points in the approximation set are distributed in the objective space as a variance:

$$SM = \sqrt{\frac{1}{|P^*|}\sum_{i=1}^{|P^*|}\left(d_i - \bar{d}\right)^2} \tag{16}$$

$$d_i = min\left\{\sum_{m=1}^{M}\left|f_m^i - f_m^k\right|\right\}_{k\in|P^*|\wedge k\neq i} \tag{17}$$

where $\bar{d}$ is the average of all d_i, M is the number of objective functions. Zero value of this metric indicates all members of Pareto front are equidistantly spaced.

5.1.3 Convergence performance metrics:

This class of performance measures quantifies the closeness of the solutions obtained in the Pareto front, evaluate how far from the non-dominated set points of each generation are in reference of the Pareto front in the final population.

Many metrics for measuring the convergence of a set of non-dominated solutions towards the Pareto front, almost all of these metrics were constructed in order to directly compare two sets of non-dominated solutions. Based on these criteria, the convergence metric selected to use in this paper is:

Generational distance (GD): This metric was proposed by [12]. GD calculates the distance of obtained Pareto optimal solutions from a selected reference set of the Pareto optimal front. Knowing that $F^{(t)}$ is the Pareto front set, of the total process, f is the objective function value and M is the number of the objective function in the problem, the mathematical formulation is as follows:

$$GD^{*} = \frac{\sum_{i=1}^{|F^{(t)}|} d_i}{|F^{(t)}|} \qquad (18)$$

Where *t* is the counter generation, F $^{(t)}$ is the Pareto front set at each generation and $\boldsymbol{d}_i$ are calculated by the follow equation:

$$d_i = min_{j=1}^{|P^*|}\left[\sqrt{\sum_{k=1}^{M}\left(\frac{f_k(i) - f_k(j)}{f_k^{max} - f_k^{min}}\right)^2}\right] \qquad (19)$$

where f_k^{max} and f_k^{min} are the maximum and minimum function values of the k^th objective function in P*. In order to keep the convergence metric within [0,1], once GD^* are calculated for all generations, these are normalized by the maximum value of them (GD^{max}) [14].

$$\overline{GD} = \frac{GD^*}{GD^{max}} \qquad (20)$$

6 The optimization methodology

The commercial code used in the multi-objective optimization process was mode FRONTIER. It is a multi-objective optimization software, which allows connecting several different CAD and/or FEA software together. Through the graphical interface is possible to set up a workflow consisting of nodes, which are connected with each other to constitute a logical scheme of an optimization process.

The numbered corresponding nodes are:

1. *Input variables:* defines design space.
2. *Design of experiment (DOE) and Scheduler:* DOE and algorithms provide different values for the input variables.
3. *Solver:* CFD or low fidelity program.
4. *Output variables:* design output variables.
5. *Objective:* minimizing or maximizing output variables.
6. *Support files.*

The figure 5 shows the input and solver Mode Frontier workflow aerodynamic problem one and the figure 6 represent the output workflow for both problems.

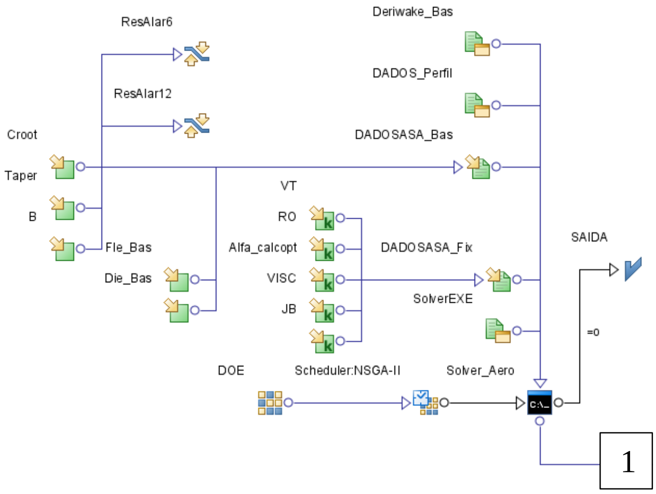


Fig.5. Input-Process Workflow (Problem 1).

In addition, the figure 7 represents the Mode Frontier input-solver workflow for aerodynamic problem 2.

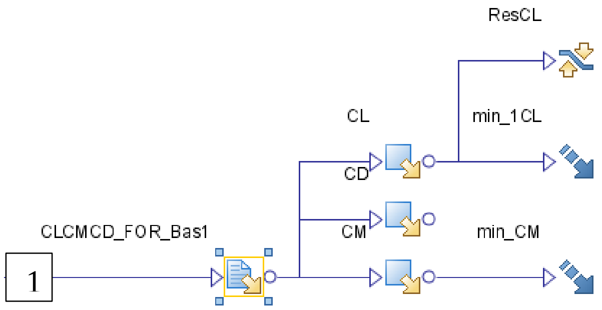


Fig.6. Output Workflow (Problem 1 and 2).

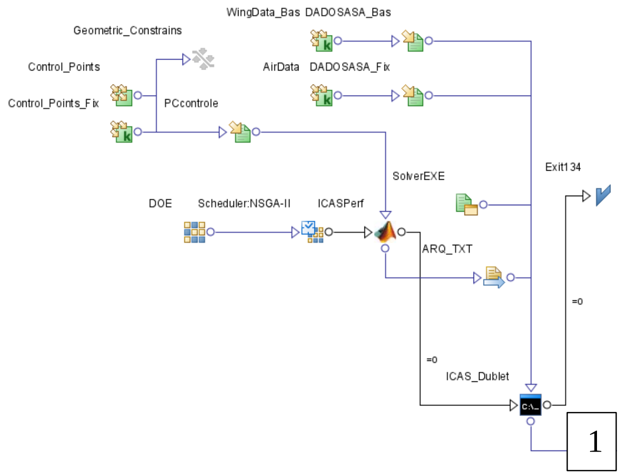


Fig.7. Input-solver Workflow (Problem 2).

6.1 Experiment Design.

In order to maintain the same initials conditions for each optimization algorithm (MOSA, MOPSO, MOGA II and NSGA II), the initial population (DOE) was maintained constant.

- Number of individuals in the initial population (*Number of designs*): 20.
- Number of iterations or *generations* 30.

A low number of generations, was taken with the purposed of reduce the computational time in each run. The total quantity of individuals was 600. In this work, all the parameter were maintained constant with the default values from Mode Frontier, with the exception of the following listed, in order to control the population size in each running.

MOSA:

- *Initial Temperature:* 1.0
- *Fraction of "Hot" Iterations:* 0.5

MOPSO:

- *Turbulence:* 0.2

MOGA II:

- *Probability of directional Cross Over:* 0.5
- *Probability of Selection:* 0.05
- *Probability of Mutation:* 0.1

NSGA II:

- *Cross Over Probability:* 0.5
- *Mutation Probability:* 0.1
- *Mutation Probability for Binary String:* 1.0

7 Results

7.1 Wing parameter variation (Problem 1)

7.1.1 Historical evolution of the population on each method.

In the figures, 8,9,10 and 11, are shown the historical evolution of the population for the algorithms: MOGAII, MOPSO, MOSA and NSGAII respectively.

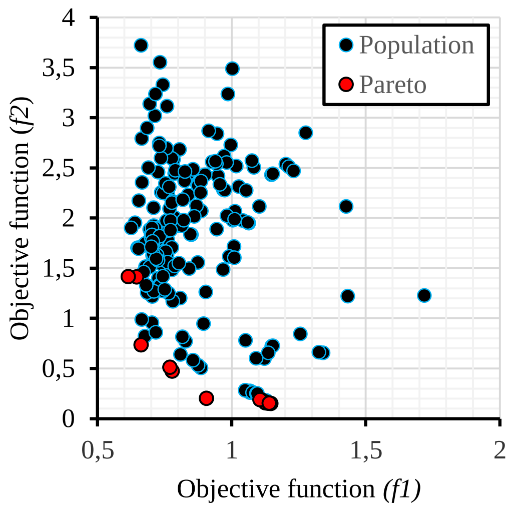


Fig.8. MOGAII Historical evolution (Problem 1).

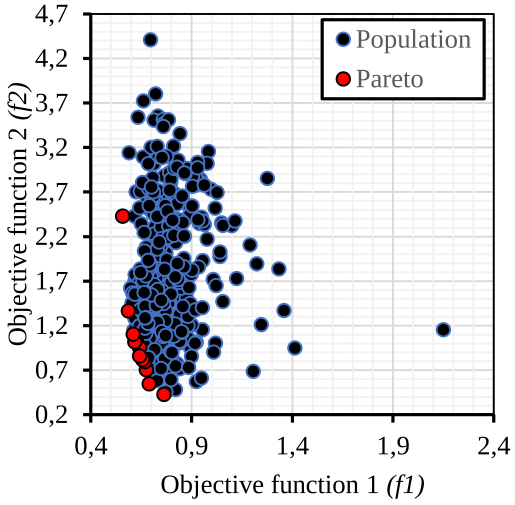


Fig.9. MOPSO Historical evolution (Problem 1).

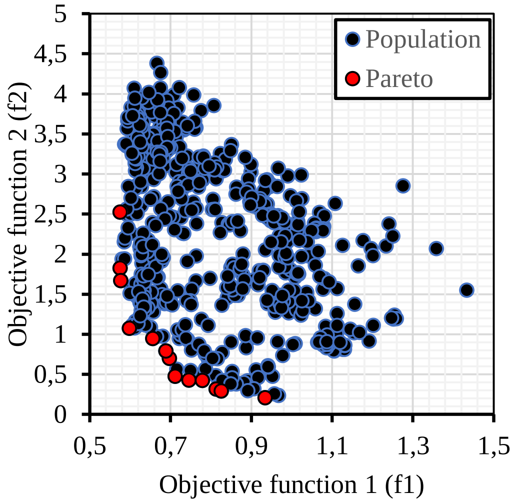


Fig.10. MOSA Historical evolution (Problem 1).

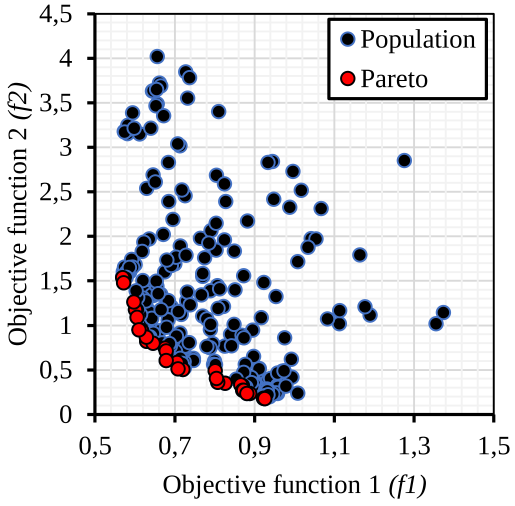


Fig.11. NSGAII Historical evolution (Problem 1).

In the figure, 8 for MOGA II algorithm can be observed that the population at the beginning is disperse and during the evolution, was difficult for the algorithm to find the Pareto front. The algorithms MOPSO and MOSA compress the population quickly, but it has a few points in his Pareto's fronts. In the figure 11 is shown that a big quantity of points in the Pareto front is obtained by a gradual and quick convergence process using the NSGA II algorithm.

7.1.2 Pareto front comparison

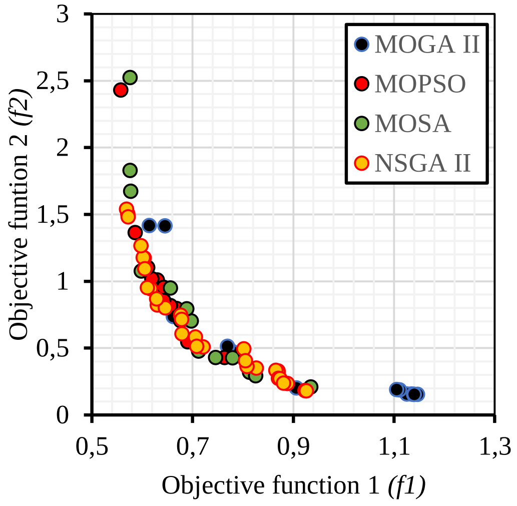


Fig.12. Pareto's front comparison (Problem 1)

In the figure 12 can be observe the Pareto's fronts of all's Pareto's fronts, for problem 1, here can be observe the dominance of the NSGA II results. The figure 8-11, show that the final data are more disperse for MOGA II and NSGA II than MOSA and MOPSO, being the last, which best compressed the population.

7.2 Airfoil wing controls point variation (Problem 2)

7.2.1 Historical evolution of the population on each method.

In the figures, 13, 14, 15 and 16, are shown the historical evolution of the population for the algorithms: MOGAII, MOPSO, MOSA and NSGAII respectively.

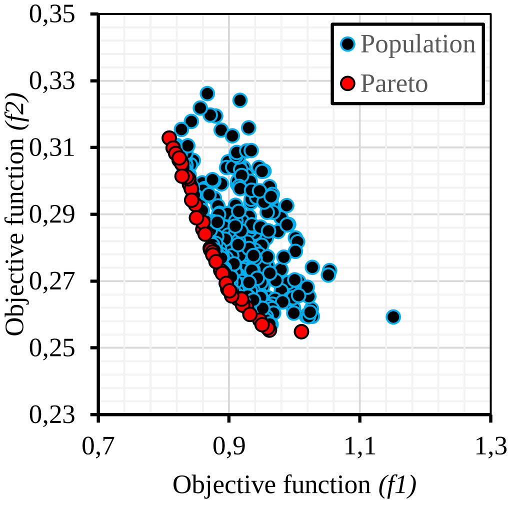


Fig.13. MOGAII Historical evolution (Problem 2).

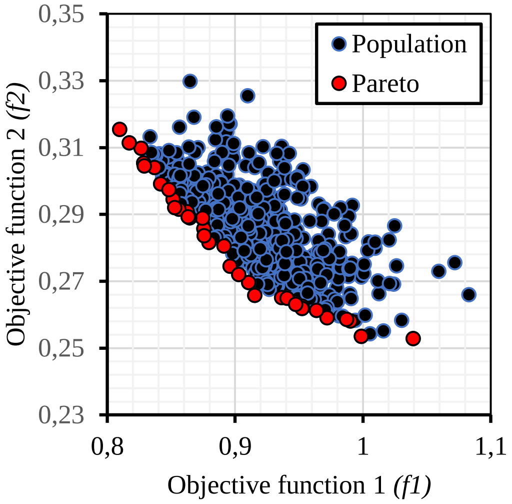


Fig.14. MOPSO Historical evolution (Problem 2).

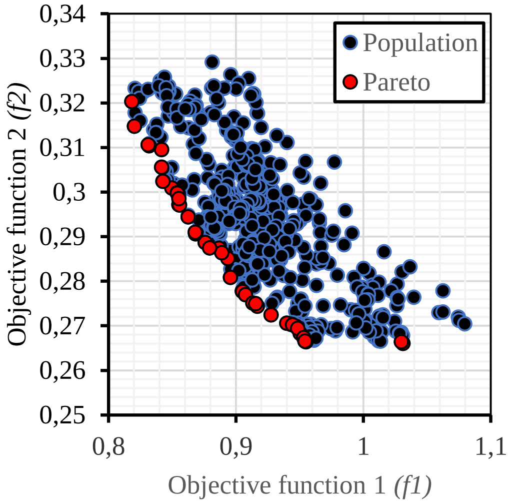


Fig.15. MOSA Historical evolution (Problem 2).

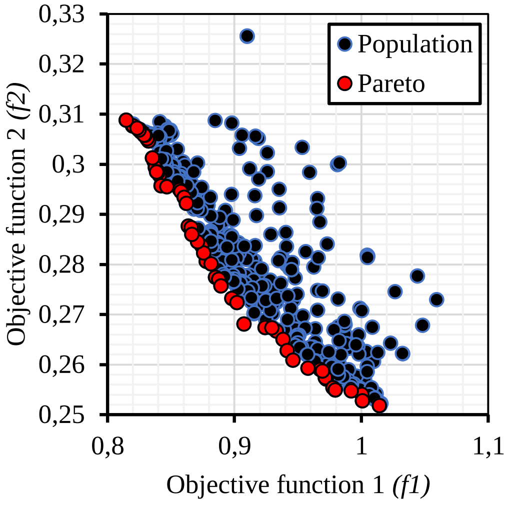


Fig.16. NSGAII Historical evolution (Problem 2).

In the figure, 16 for NSGA II algorithm can be observed that the population at the beginning is disperse and during the evolution, was very easy for the algorithm to find the Pareto front. The algorithms MOPSO and MOSA compress the population quickly, attains a lot of points in his Pareto's fronts. In the figure 15 is shown that an instable and quick convergence process using the MOGA II algorithm obtains a big quantity of points in the Pareto front.

7.2.2 Pareto front comparison

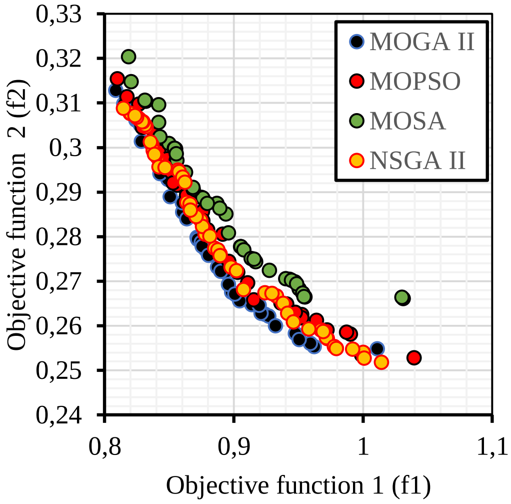


Fig.17. Pareto's front comparison (Problem 2)

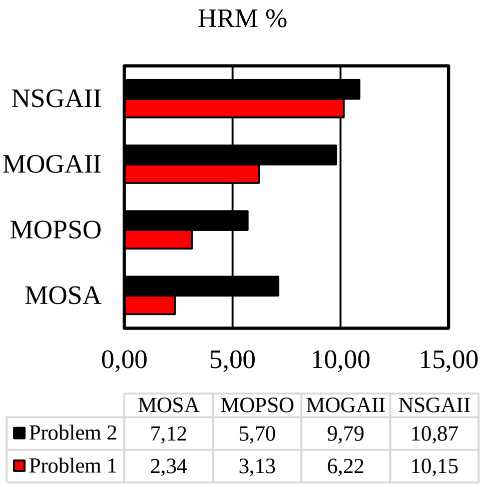
In the figure 17 can be observed the Pareto's fronts of all's Pareto's fronts, in the problem 2, it can observed the dominance of the MOGA II results. The figure 13-16, show that the final data are being compressed in all method, but the best behavior is obtained with MOGAII and NSGA II.

7.3 Performance metrics results.

In order to calculate the Multi Objective Performance Metrics (MOPM), here was used a MATLAB script created, based on the theoretical methodology described in the section 5, to obtain the values for each method.

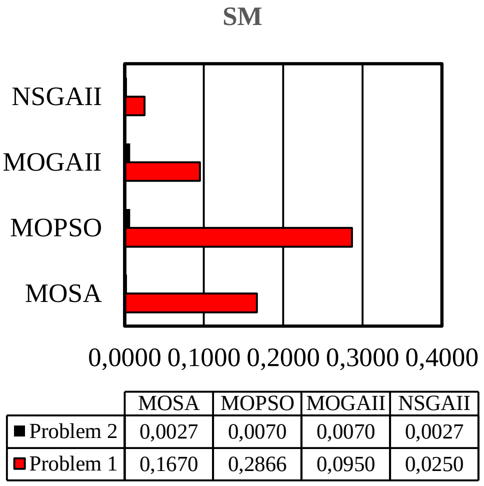
7.3.1 HRM and SM Results

The figure 18 and 19 show the values for HRM and SM for each multi objective optimization method applied to the aerodynamic problems 1 and 2.



	MOSA	MOPSO	MOGAII	NSGAII
■ Problem 2	7,12	5,70	9,79	10,87
🟥 Problem 1	2,34	3,13	6,22	10,15

Fig.18. Hit Rate Metrics (HRM).



	MOSA	MOPSO	MOGAII	NSGAII
■ Problem 2	0,0027	0,0070	0,0070	0,0027
🟥 Problem 1	0,1670	0,2866	0,0950	0,0250

Fig.19. Spacing Metric (SM).

The HRM values obtained, show that NSGA II has a good capacity to find a big number of points on the Pareto front, for both aerodynamics wing problems. The MOSA has the worst capacity for the problem 1 and MOPSO for the problem 2.

The figure 17 show that the space between the Pareto points are uniformly distributed for both problems with NSGA II and the worst values for this parameter were calculated for MOPSO.

7.3.2 GD Results

The figure 20, 21, 22 and 23 show the graphics GD vs. Generations for each multi objective optimization method applied to the aerodynamic problems 1 and 2.

The graphics for GD show that the best convergence for both aerodynamics wing problems is obtained using the NSGA II method. The convergence is not attain using MOPSO and MOGA II in the problem 1. MOGA II, reached the convergence for the problem 2 but not for the problem 1. MOSA has the longer and instable convergence in both problems. Is good to take in account that in this paper the number of generation fixed in each method was low in order to reduce the computational time.

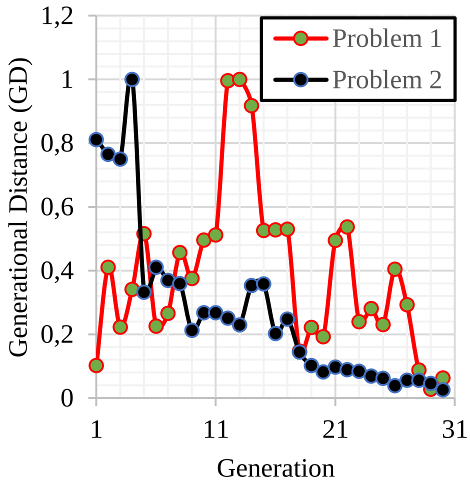


Fig.20. GD Vs Generation (MOGA II).

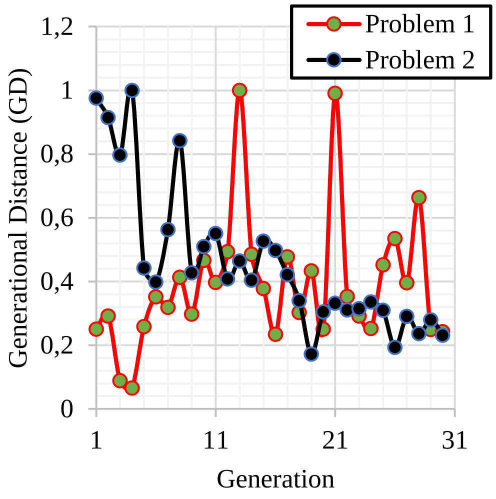


Fig.21. GD Vs Generation (MOPSO).

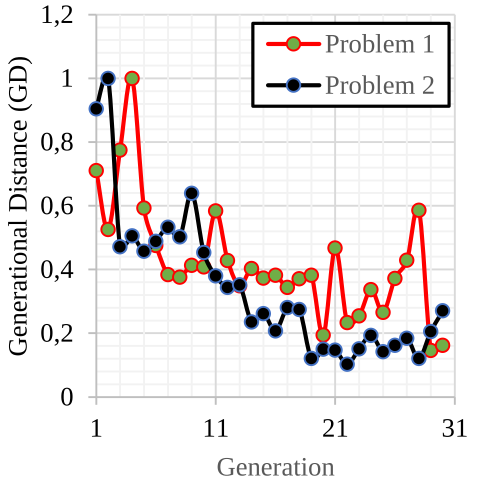


Fig.22. GD Vs Generation (MOSA).

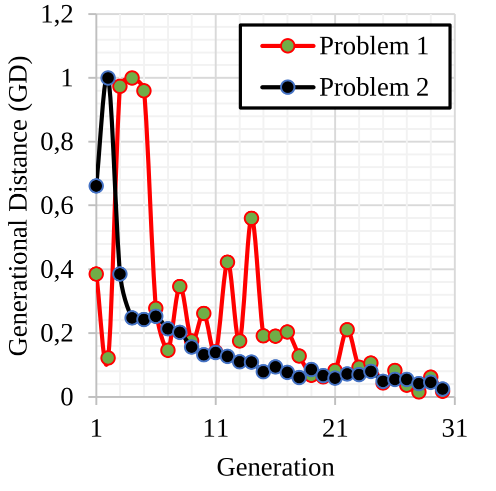
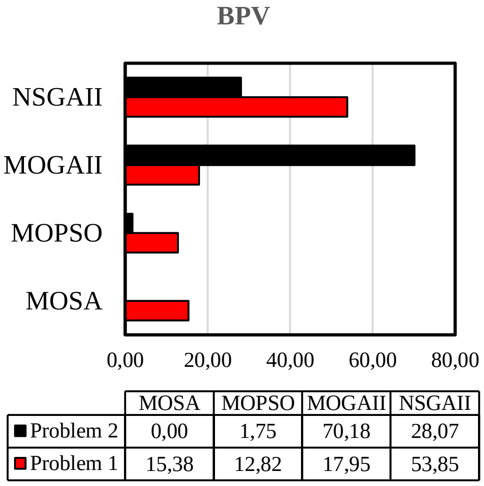


Fig.23. GD Vs Generation (NSGA II).

8 Conclusions.

This paper does a comparison between different Multi Objective Optimization Algorithm (MOOA), using three performance metrics for measuring capacity (HRM), coverage (SM) and convergence (GD) of the MOOA, applied on two aerodynamics wing problems. In each problem, the objective functions are calculated through the black-box computational code based on first order 3D panel method, for lift, moment and induced drag coefficient calculation, and the boundary layer method for the calculation of the friction drag coefficient. The performance metrics were used to quantitatively evaluate and compare four popular stochastic multi objective optimization methods, NSGA-II, MOSA, MOPSO and MOGA-II, in order to analyze the behavior of each algorithm in the aerodynamics wing design application.

In the direct application of each MOOA was constructed, the Pareto’s fronts of all’s Pareto’s fronts, in this case the best algorithm for the problem 1 that involve only wing parameters, was NSGA II with a contribution of 53,85% of the total points in the general Pareto. It means that the major quantity of points of NSGA II Pareto front, dominate to the Pareto points of others MOOA analyzed. The same analysis was did it for the problem 2, but in this case the best algorithm was MOGA II with a 70,18 %. This is shown in the figure 24.



	MOSA	MOPSO	MOGAII	NSGAII
■ Problem 2	0,00	1,75	70,18	28,07
🟥 Problem 1	15,38	12,82	17,95	53,85

Fig.24. Best Pareto Values

The HRM values obtained, show that NSGA II has a good capacity to meet a big number of points on the Pareto front for both aerodynamics wing problems. It means that with NSGA II have obtained a big quantity of solutions in the Pareto front.

With respect to the SM values it can be concluded that the quality of the NSGA II Pareto front for both problems was the best, due to this metric value is low, when the space between the Pareto points decrease.

GD graphics show that the best convergence for both aerodynamics wing problems is obtained using the NSGA II method. Now we have to take into account that in this paper the number of generation fixed in each method was low in order to reduce the computational time, these results may change if each algorithm is allowed to reach more generations, but in this case we can conclude that NSGA II has a quickly converge. In order to reduce the cost computational due to use the CFD, in future works, will be introducing the metamodelagem techniques.

8 References.

[1] N. Diaz, N. Manzanares, and B. Sousa, "Metodologia para diseño aerodinamico de alas basado en tecnicas de optimizacion multiobjetivo," in *CIBEM 2017*, 2017.

[2] K. Deb, “Multi-Objective Optimization Using Evolutionary Algorithms,” *John Wiley & sons, LTD.* p. 497, 2001.

[3] C. S. Tsou, “Multi-objective inventory planning using MOPSO and TOPSIS,” *Expert Syst. Appl.*, vol. 35, no. 1–2, pp. 136–142, 2008.

[4] R. C. Eberthart and Yuhui Shi, *Computational Intelligence*, First. Burlington: Morgan Kaufmann, 2007.

[5] V. Kumar, “Multi-Objective Particle Swarm Optimization: An Introduction,” *Smart Comput. Rev.*, vol. 4, no. 5, 2014.

[6] G. Meneghetti, V. Pediroda, and C. Poloni, “Application of a Multi Objective Genetic Algorithm and a Neural Network to the optimisation of foundry processes .,” *Evol. Algorithms Eng. Comput. Sci.*, no. January 2015, pp. 457–470, 1999.

[7] C. M. Fonseca and P. J. Fleming, “Multiobjective optimization and multiple constraint handling with evolutionary algorithms - Part I: A unified formulation,” *IEEE Trans. Syst. Man, Cybern. Part ASystems Humans.*, 1998.

[8] E. Zitzler, M. Laumanns, and L. Thiele, “SPEA2: Improving the Strength Pareto Evolutionary Algorithm,” *ETH Zurich Res. Collect.*, vol. 1, pp. 1–20, 2001.

[9] S. Poles, E. Rigoni, and T. Robic, “Bioinspired optimization methods and their applications,” in *MOGA-II performance on noisy optimization problems*, 2008, no. October, pp. 51–62.

[10] B. Sousa, “Projeto de Perfis Aerodinâmicos Utilizando Técnicas de Otimização Mono e Multiobjetivos.” Itajuba, p. 131, 2008.

[11] E. Zitzler, K. Deb, and L. Thiele, “Comparison of Multiobjective Evolutionary Algorithms: Empirical Results,” *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, 2013.

[12] D. A. Van Veldhuizen and G. B. Lamont, “On Measuring Multiobjective Evolutionary Algorithm Performance,” *Proc. IEEE Congr. Evol. Comput.*, vol. 1, pp. 204–211, 2000.

[13] J. Schott, “Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization.,” *Massachusetts Inst. Technol. Cambridge, Massachusetts*, 1995.

[14] K. Deb and S. Jain, “Running Performance Metrics for Evolutionary Multi-Objective Optimization,” in *Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning (SEAL’02)*, 2002, vol. 1, pp. 13–20.

Copyright Statement