

# COMPOSITE OPTIMIZATION PERFORMANCE COMPARISON BETWEEN GENESIS<sup>®</sup>/ BIGDOT AND MODEFRONTIER<sup>®</sup>'S MOGAI ALGORITHMS

Saullo Giovanni Pereira Castro, saullogiovani@gmail.com<sup>1</sup>

Thiago Augusto Machado Guimarães, thiagoamg@gmail.com<sup>1</sup>

José Antônio Hernandes, hernandes@ita.br<sup>1</sup>

<sup>1</sup>Instituto Tecnológico de Aeronáutica endereço para correspondência - ITA, Praça Marechal Eduardo Gomes, 50 - Vila das Acácias - CEP 12.228-900 - São José dos Campos - SP - Brasil

**Abstract.** *The purpose of this study is to evaluate modeFRONTIER<sup>®</sup>'s MOGAI and GENESIS<sup>®</sup>'s BIGDOT performances on the problem of optimizing a composite wing under two load cases, 3.0G up and 1.0G down cruise loads. The minimization of the structural weight was the objective under the constraint of keeping the buckling load factor eigenvalue higher than “1.00”. The convergence pattern and achieved results were compared in order to evaluate the pros and cons of each one. The vast sample space involved in this optimization that applies 144 variables at three levels each is a real challenge. It is expected from the given solutions to provide not only light structures but feasible ones, with reasonable stacking sequences.*

**Key-words:** *composite optimization, stacking sequence, genetic algorithm, GENESIS, modeFRONTIER*

## 1. INTRODUCTION

### 1.1 Composite materials and optimization

The use of composite materials as primary structures in the commercial aviation industry has been gradually increasing over the last decade. This has culminated in programs such as the Airbus A350 or the Boeing 787, for which composite materials will play a major role (Herencia *et al.*, 2007). In general, composite materials present high specific strength and stiffness ratios (Jones, 1999). Because of practical, yet often limiting, manufacturing considerations, laminated fiber composite panels have been restricted to symmetric or midplane symmetric laminates with 0, 90, 45 or -45 deg ply angles (Herencia *et al.*, 2007).

Composite materials are becoming the material of choice in structural applications today and for the future. Although the high strength-to-weight properties of composite materials are attractive, their greatest advantage is to provide the designer the ability to tailor directional strength and stiffnesses of a material to a given loading environment of the structure (Jones, 1999).

The advantages of using composite materials are somewhat offset by the more complex structural analysis that is required. Composites usually involves large, non-convex, integer programming problems that are discrete in nature. Genetic algorithms (GAs) are ideal optimization algorithms for these type of problems and are simple to implement when compared to other optimization techniques (Soremekun, 1997). However, if the cost to analyze each structure is high then a designer may be forced to use other optimization schemes rather than GAs (Soremekun, 1997), since its associated computational cost is usually comparatively high (Herencia *et al.* (2007) and Haftka and Walsh (1992) and Castro (2009) and Nagendra *et al.* (1996) and Schimit and Farshi (1973) and Soremekun (1997)). Castro (2009) has showed some strategies and guidelines to reduce computational cost of a GA driven optimization. It is expected from a laminate optimized using a GA to reach lower structural weight for the same objective and constraint set than many other algorithms such as simulated annealing or “SIMPLEX” gradient method (Castro, 2009).

Genetic algorithms (GAs) apply techniques for exploring a given sample space that allows the designer to take advantage of an unbalanced and an unsymmetric laminate for a given application. Commonly, composite design applies symmetry and balance in order to simplify the analysis since unsymmetric and unbalanced composites show a stress-strain behavior too difficult to predict (Jones (1999) and Tsai (2008)). All laminates used in this study are symmetric and have balanced plies, i.e. each ply that is off-axis has a corresponding ply with the same angle with a negative sign, i.e. a ply with -30° will have a corresponding +30°, or a ply with -45° will have a corresponding with +45°.

Castro (2009) proposed a methodology for generating the laminates that uses a fixed stacking sequence and the input variables are the number of plies for each orientation that is to be used in the laminate. Guimaraes *et al.* (2010) developed a methodology in which a fixed template for a laminate is used. Into the fixed blanks the thickness of each ply and the corresponding angle is inputted. The thickness may vary from a small value that represents a zero thickness to an integer multiple of the thickness of a single ply, passing through discrete values. The angles vary accordingly to the list of discrete angles to be used in the laminate. This way of optimization doesn't restrict the stacking sequence, allowing the optimization to reach better results. Also, the sample space using Guimaraes' methodology is significantly smaller than the sample space necessary for Castro's methodology.

In the present study Guimaraes' methodology was used to achieve the optimization of a composite wing applying two commercial softwares that use different algorithms. First the modeFRONTIER<sup>®</sup>'s MOGAII - Multi Objective Genetic Algorithm (ESTECO, 2003). Second the GENESIS<sup>®</sup>'s BIGDOT algorithm designed for very large scale optimization problems. For discrete variable problems there is no guarantee of obtaining the true minimum (VRD, 2010).

The objective is to minimize the structural weight under the constraint of having the linear buckling load factor higher than "1.00". These proved to work well on finding a best laminate for a given application (Guimaraes *et al.* (2010) and Castro (2009)), especially in applications without concentrated loads where the fail index might also be necessary. In the case of large panels using distributed loads the instability criteria is usually more critical than static stresses (Bruhn, 1965).

Although many algorithms have been compared to GAs (Castro, 2009), the BIGDOT performance has not been compared. As GENESIS<sup>®</sup> is vastly applied for structural optimization, it is important to know how well are its results for discrete optimization, by comparing with already proved methods, such as GAs.

Genetic algorithms are less likely to be trapped in local optimum design areas (Soremekun, 1997). Both Goldberg (1989) and Buckles and Petry (1992) have noted several distinct differences between GAs and classical optimization algorithms. First, GAs work with objective functions, not relying only on derivative information that may lead to local minimums when dealing with non-smooth functions (Soremekun, 1997). Second, GAs may work with groups of designs, called populations, searching many areas of the sample space at once. For this reason GAs are implicitly parallel algorithms. Finally, genetic algorithms rely on a set of probabilistic rules instead of deterministic ones. This randomness in the search process guarantee its robustness (Soremekun, 1997).

## 1.2 Software modeFRONTIER<sup>®</sup>

The modeFRONTIER<sup>®</sup> has been designed and distributed by ESTECO. It is mainly a mathematical software designed for optimization. Into the optimization process it integrates any software that works in batch mode with an input and an output file, like many solvers and most common engineering applications. This integration capability has put modeFRONTIER<sup>®</sup> in a large range of applications (ESTECO, 2003).

The algorithm selected for this study is the MOGAII - Multi Objective Genetic Algorithm II. The version II includes directional cross-over. While the conventional cross over considers data from parents only, the directional cross-over also takes into account data from grand-parents. In the directional cross over values for all variables to the new individuals are calculated using conventional cross-over adding an increment that depends on the grandparents' variables and fitness functions. Equation (1) shows how a new variable X is calculated [modeFRONTIER<sup>®</sup> user manual].

$$X = X_i + S * \text{signal}(A_i - A_{i_1}) * (X_i - X_{i_1}) + T * \text{signal}(A_i - A_{i_2}) * (X_i - X_{i_2}) \quad (1)$$

The "X<sub>i</sub>" value is taken from the conventional cross over operation. This value comes from a parent identified by "i". Two grandparents have generated the parent "i". These are called "i<sub>1</sub>" and "i<sub>2</sub>". The terms multiplied by "S" and "T" are the increments. "A<sub>i</sub>", "A<sub>i<sub>1</sub></sub>", "A<sub>i<sub>2</sub></sub>" are the fitness functions. As it can be seen the increments assume either a positive or negative value, depending on the fitness functions values. This allows the directional increments to drive the new values for the variable "X" towards the maximization of the fitness function.

Despite directional cross over, all other operations are more similar to most common genetic algorithms. Selection keeps best individuals for next generations and mutation selects some individuals to randomly change its variables. The number of individuals chosen for selection and mutation is determined by "Probability of Selection - Psel" and "Probability of Mutation - Pmut", respectively. If "Elitism" parameter is set as "Enabled" then the individuals chosen by selection operation will be kept with the exactly values for all variables to the next generation. The "Psel" parameter must ensure at least one individual lasting for the next generation, otherwise the good results found along the optimization will be predictably lost. To keep at least one best individual is enough but not advisable because any problem with the run of this optimal individual may lead to the loss of a best result, that can be found again after many generations, increasing the optimization cost (Castro, 2009).

The individuals chosen for mutation will have their "DNA" changed at a some level that is set by the "DNA string mutation ratio - DNAsmr" parameter. An illustration of how does this string mutation occurs is showed in Fig. (1) taken from Castro (2009).

Individual String	110010101101110					total of 15 strings
	VAR1	VAR2	VAR3	VAR4	VAR5	
Values	12	10	6	3	2	
Bynary Code	1100	1010	110	11	10	

String mutation ratio of 30% (5 strings will be changed)						
New string	11100011001110					
	VAR1	VAR2	VAR3	VAR4	VAR5	
New bynary code	1110	0001	100	11	10	
New values	14	1	4	3	2	

String mutation ratio of 60% (10 strings will be changed)						
New string	111100010011011					
	VAR1	VAR2	VAR3	VAR4	VAR5	
New bynary code	1111	0001	001	10	11	
New values	15	1	1	2	3	

Figure 1. Mutation operation and how the DNA string mutation ratio affects new individuals.

It is important to keep in mind that the directional cross over and selection operations show more assured performance while mutation usually takes longer turns to produce improvements towards the objective, since it involves random operations. By reducing the string mutation ratio it is possible to control at some level the randomness of this process, as showed in Fig. (1).

One may choose small values for this parameter in order to ensure faster convergence. The problem is that the robustness of the method largely rely on the “DNAsmr” parameter, which guarantees the creation of new individuals in different regions of the sample space, keeping a well likelihood for finding new optimum sets for the variables of the individuals (Castro, 2009).

The importance of the “DNAsmr” parameter increases when a small population size is used. Small populations may lead to efficient optimizations since fewer runs will be necessary before evolving the population. The issue is that the amount of available genetic code when using small populations is comparatively shrunken reducing the possibilities for new crossed-over individuals. The usage of “DNAsmr” between 5% and 10% and a larger number of generations for small populations guarantees good efficiency and robustness (ESTECO (2003) and Castro (2009)). It is better to use smaller population sizes with more generations than setting larger populations and fewer generations (Castro, 2009). The optimum value for the “Pmut” parameter presumably increases when using smaller populations. Similarly to “Psel” parameter, the “Pmut” must guarantee that at least one individual is chosen for mutation, e.g. using a population of 5 individuals need this parameter set to a minimum of 20%, a population of 10 will need this parameter set to a minimum of 10% and so on (Castro, 2009).

### 1.3 Software GENESIS<sup>®</sup>

Genesis is a complete commercial optimization software. It’s DOT and BIGDOT algorithms are the options to perform a structural optimization.

In accordance with Vanderplaats Research & Development, Inc. DOT is a general-purpose gradient-based optimization software library that can be used to solve a wide variety of optimization problems. Users have to link the DOT library into their own program. DOT provides the optimization technology, while the rest of the program has to provide the required function evaluations needed to perform the optimization. These function evaluations can be linear or nonlinear functions of the design variables. They may be very simple analytical functions or may be highly complicated implicit functions, for example a non-linear structural finite element simulation. Very little formal knowledge of optimization techniques is needed to make efficient use of DOT. This algorithm can handle constrained, unconstrained, linear and non-linear optimization problems and can automatically calculate finite difference gradients needed during the optimization. DOT can also deal with user supplied gradients.

The BIGDOT is a gradient-based optimization library, similar to DOT. Users link the BIGDOT library with their program to perform optimization. BIGDOT is intended to solve very large scale optimization problems and can handle linear, non linear, constrained and unconstrained optimization problems. BIGDOT is capable of solving optimization problems with hundreds of thousands of design variables and can handle problems with large numbers of active constraints, up to the number of design variables. In addition discrete/integer and mixed variable problems can also be solved. For discrete variable problems, there is no guarantee of obtaining the true minimum. The techniques in BIGDOT are intended to efficiently obtain a near optimum discrete solution for large scale optimization problems.

## 2. PROPOSED METHODOLOGY

### 2.1 The FEM model and optimization variables

The finite element model of the composite wing is showed in Fig. (3), originally used by Castro (2009). Two load cases, 3.0G up and 1.0G down, were evaluated to perform a kind of simple but representative analysis. The load application was done using “PLOAD4” that uniformly distributed the airplane total weight as applied pressure. The airplane weight is 100000 lb, or 45360 kg. For each semi-wing the supported weight is approximately the half of the total weight, i.e. 50000 lb or 22680 kg. For the 3.0G up load case the pressure was distributed in the wing lower skin, that has a total wet area of 25.81 m<sup>2</sup>. In the 1.0G down load case the pressure was applied in the lower skin, with a total wet area of 25.63 m<sup>2</sup>. Inertial loads were not applied to the wing and the fuel into the wing. These loads can significantly influence the analysis, so they should be considered for a more detailed study where the design itself is the main focus. To the comparative study presented herein the equivalent aerodynamic load using the maximum take-off weight as a basis is adequate. The same loading method was used both in modeFROTIER<sup>®</sup> and GENESIS<sup>®</sup>.

The used input card for laminate composites is the “PCOMP”, showed in Tab. (1). The basic template of all cards used in this study is showed in the Tab. (1). Each ply is defined with: material (MID), thickness (tXX) and angle ( $\theta$ XX). The option “SOUT” is also necessary for each ply to inform the solver whether or not the stress output values will be printed for the corresponding ply (MSC, 2008).

Table 1. Typical PCOMP card.

PCOMP	PID	Z0	NMS	SB	FT	TREF	GE	LAM
	MID <sub>1</sub>	t <sub>1</sub>	$\theta$ <sub>1</sub>	SOUT <sub>1</sub>	MID <sub>2</sub>	t <sub>2</sub>	$\theta$ <sub>2</sub>	SOUT <sub>2</sub>
	MID <sub>3</sub>	t <sub>3</sub>	$\theta$ <sub>3</sub>	SOUT <sub>4</sub>	MID <sub>4</sub>	t <sub>4</sub>	$\theta$ <sub>4</sub>	SOUT <sub>4</sub>
	...	...	...	...	...	...	...	...
	MID <sub>N-1</sub>	t <sub>N-1</sub>	$\theta$ <sub>N-1</sub>	SOUT <sub>N-1</sub>	MID <sub>N</sub>	t <sub>N</sub>	$\theta$ <sub>N</sub>	SOUT <sub>N</sub>

- PID: PCOMP propertie ID;
- Z0: Distance from the reference plane to the bottom surface. Default value used is “-0.5 times the laminate thickness”;
- NSM: Non structural mass per area unit. Default value used is “0.1908”;
- SB: Bound shear allowable. Default value used is “1.31E+07”;
- FT: Failure theory applied in the analysis. Default theory used is “HILL”;
- TREF: Reference temperature. Default value used is “298.15”;
- GE: Damping coefficient. Default value used id “0.0”;
- LAM: Laminate options. Default option used is “SYM” to create a symmetric laminate;
- MID: Ply material ID. Default used is “1”;
- tXX: ply thickness;
- $\theta$ XX: ply angle;
- SOUT: output option for plies. Default value used is “YES”.

To guarantee a symmetric laminate the option LAM = “SYM” is used. In this way all the defined plies are half of the laminate. If an odd number of plies is desired for the full laminate, the center ply thickness should be half of the actual thickness (MSC, 2008).

Not only symmetric but also balanced laminates were used. Guimaraes *et al.* (2010) applied a simple and useful methodology to produce balanced laminates. This ensures reduction of laminate terms related to bending-twisting coupling (Tsai (2008) and Jones (1999)), as showed in Fig. (2). The balancing was done by adding an opposite-angle ply for each ply off-axis. In Tab. (2) the negative entries “- $\theta$ XX” on the right are used as balancing plies. They have plies with the same thickness and opposite angles when comparing with the plies on the left.

The frontal and rear spar, as well as the lower and upper skin have their laminates reinforced from the tip to the root, to accomplish the load increase towards this same direction. In Fig. (3) the upper skin laminates are named. Note that properties identification follows a pattern. In this pattern the laminate “206” is the reinforced “205”, that is the reinforce “204” and so on, until the base laminate “200”. The base laminate is always closer to the wing tip.

$$\begin{array}{c}
 \left\{ \begin{array}{l} N_x \\ N_y \\ N_{xy} \end{array} \right\} = \left[ \begin{array}{ccc} A_{11} & A_{12} & A_{16} \\ A_{12} & A_{22} & A_{26} \\ A_{16} & A_{26} & A_{66} \end{array} \right] \left\{ \begin{array}{l} \epsilon_x^0 \\ \epsilon_y^0 \\ \gamma_{xy}^0 \end{array} \right\} + \left[ \begin{array}{ccc} B_{11} & B_{12} & B_{16} \\ B_{12} & B_{22} & B_{26} \\ B_{16} & B_{26} & B_{66} \end{array} \right] \left\{ \begin{array}{l} \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{array} \right\} \\
 \left\{ \begin{array}{l} M_x \\ M_y \\ M_{xy} \end{array} \right\} = \left[ \begin{array}{ccc} B_{11} & B_{12} & B_{16} \\ B_{12} & B_{22} & B_{26} \\ B_{16} & B_{26} & B_{66} \end{array} \right] \left\{ \begin{array}{l} \epsilon_x^0 \\ \epsilon_y^0 \\ \gamma_{xy}^0 \end{array} \right\} + \left[ \begin{array}{ccc} D_{11} & D_{12} & D_{16} \\ D_{12} & D_{22} & D_{26} \\ D_{16} & D_{26} & D_{66} \end{array} \right] \left\{ \begin{array}{l} \kappa_x \\ \kappa_y \\ \kappa_{xy} \end{array} \right\}
 \end{array}$$

SHEAR-EXTENSION COUPLING      BENDING-EXTENSION COUPLING  
 BENDING-EXTENSION COUPLING      BEND-TWIST COUPLING

Figure 2. Taken from Jones (1999). Physical significance of stiffness terms in force and moments resultants.

In Fig. (3) it is also showed the maximum number of plies that the upper skin laminates may achieve. Despite laminate “200” has only 14 thicknesses variables, these variables can assume values from “0\*t” to “2 \* t”, where “t” is the nominal thickness of a single carbon tape ply. Each ply is mirrored twice in order to give a balanced and symmetric laminate, as further explained below. Due to this when a laminate has 14 thicknesses variables, it may have “14 \* 2 \* 2 \* 2” plies (“2\*t”, 1 mirror due to balance, 1 mirror due to symmetry). Applying this same rule for the reinforcing laminates “201”, “202”, “203”, “204”, “205”, “206”, for each thickness variable included, a maximum of 8 plies can be added. The reinforcing laminates carries all plies informations from the later laminate, plus the additional plies corresponding to its own “tXX” and “θXX” variables.

All thicknesses and angles are discrete variables. For a laminate built of a single material, the resulting thickness of the laminate must be an integer multiple of the thickness of a single ply, called “t”. Despite all plies could be stacked in any continuous angle when using automated processes such as automatic tape laying, the use of discrete values makes the laminate easier to manufacture (Jones (1999), Herencia *et al.* (2007) and Tsai (2008)) and the sample space smaller (Castro, 2009).

Table 2. PCOMP card configured to Guimaraes’ methodology.

PCOMP	PID	Z0	NMS	SB	FT	TREF	GE	LAM
	MID	Vt <sub>1</sub>	θ <sub>1</sub>	SOUT	MID	Vt <sub>1</sub>	-θ <sub>1</sub>	SOUT
	MID	Vt <sub>2</sub>	θ <sub>2</sub>	SOUT	MID	Vt <sub>2</sub>	-θ <sub>2</sub>	SOUT
	MID	Vt <sub>N</sub>	θ <sub>N</sub>	SOUT	MID	Vt <sub>N</sub>	-θ <sub>N</sub>	SOUT

By varying the values of “tXX” the number of plies is indirectly chosen. As the plies on the right are mirrors of the plies on the left and the laminate is mirrored as a whole by using the “SYM” option, an increase in the variable “tXX” in one thickness results in the addition of 4 plies in the laminate, due to the two mirrors. This fact lets the optimization process to a sensibility level of 4 plies, i.e. the optimization will add or remove plies in groups of four.

The corresponding angle “θXX” only assigns to these thickness variable an orientation. As all plies are mirrored twice, any angle value can be assigned in order to keep a balanced and symmetric laminate.

These design variables take their values from lists with discrete values. For thickness the list is [0, t, 2\*t]. For angle the list is [0, 45, 90]. The mirroring operation automatically creates the opposite angles [-0, -45 and -90]. Note that only for the +45 and -45 pair the mirroring causes some effect. The stacking sequence will be a consequence of the optimal choice of thickness and angles, since the thicknesses can be eliminated assuming the zero (approximately) value.

Two steps were followed in the implementation of the proposed strategy. First, an initial analysis shall be done in order to generate a laminate that holds the applied loads and accomplish the applied constraints. Now, with a correct “PCOMP” template it is possible to create a compatible number blanks to be filled with all thicknesses and angles variables. This initial process avoids unnecessary design variables, lowering the computational cost of the optimization.

The second step is the correct association of thicknesses and angles fields to the respective thickness and angle design variables, as showed in Tab. (2).

After the optimization is finished the obtained laminate has many dummy layers, i.e. layers with thicknesses close to zero. Table (3) shows this situation. The removal of these layers is part of the optimization process and doesn’t change the analysis results since their very small thicknesses don’t influence significantly the results. The existance of these plies in the final results simply means that they aren’t necessary.

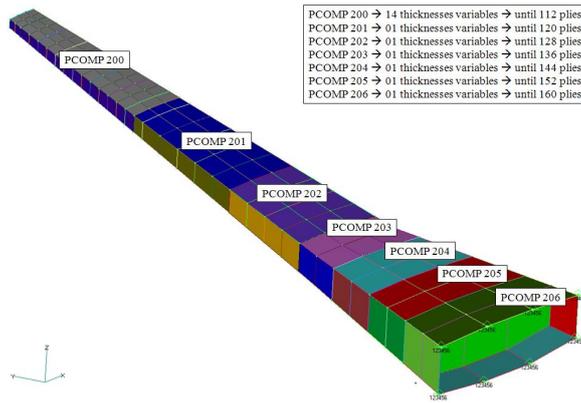


Figure 3. FEM model used in the optimizations.

Table 3. Removal of dummy plies with thicknesses near zero.

PCOMP	PID	Z0	NMS	SB	FT	TREF	GE	LAM
	MID	1.8000-4	0.	SOUT	MID	1.8000-4	-0.	SOUT
	MID	3.6000-4	45.	SOUT	MID	3.6000-4	-45.	SOUT
	MID	1.8000-4	90.	SOUT	MID	1.8000-4	-90.	SOUT
	MID	1.8000-8	90.	SOUT	MID	1.8000-8	-90.	SOUT
	MID	1.8000-8	45.	SOUT	MID	1.8000-8	-45.	SOUT
	MID	3.6000-4	0.	SOUT	MID	3.6000-4	-0.	SOUT
	MID	3.6000-4	0.	SOUT	MID	3.6000-4	-0.	SOUT
	MID	3.6000-8	45.	SOUT	MID	3.6000-8	-45.	SOUT

## 2.2 The sample space

The total number of variables involved in this study is showed in Tab. (4). The columns show the total number of thicknesses and angles variables, that are essentially the same when using Guimaraes' methodology (Guimaraes *et al.*, 2010). There are 72 thicknesses variables plus 72 angles variables.

The number of possible configurations for the laminates is the size of thickness list (“t\_base”, that is in this case 3 since the thickness may assume 0 or 1\*t or 2\*t) multiplied by the size of angle list (“θ\_base”, that is 3 in this case since the angle may be 0 or 45 or 90) powered by the number of pairs of thickness and angle added by a group. Equation (2) shows this calculation. In Eq. (2) “n = 72”, since it refer to the number of thicknesses variables or angles variables. The sample space calculated for this study is  $(3*3)^{72} = 5.1*10^{68}$  possible laminates combinations. Such sample space is impossible to be completely tested in order to find the best result. It is really challenging for optimization algorithms dealing with such a vast space.

$$SE = (t\_base)^{n_{thickness}} * (\theta\_base)^{n_{angle}} = (t\_base * \theta\_base)^n \tag{2}$$

It is important to emphasize that the laminates “200”, “300”, “400” and “500” works as base laminates for the others. The number of plies added to these base laminates is given in Tab. (4). For example, laminate “301” is “300” plus additional plies, “302” is “301” plus additional plies and so on. For each propertie identification (PID) the number of extra plies in the reinforcing laminate is showed. Please refer to Fig. (3) to see how the laminates are reinforced from the tip to the root, and note the propertie identification pattern that was adopted.

## 2.3 MOGAIL configuration

The Multi Objective Genetic Algorithm, MOGAIL, was configured with the parameters shown in Tab. (5). “Random Seed” when specified guarantees for the optimization the same random pattern, i.e. even the random events of the optimization such as: the choice of individuals for cross-over, mutation and selection, DNA string mutation and any other; may be reproduced if the user wants to restart an optimization, obtaining then the same results. “Concurrent Runs” allows the user to perform parallel processing if more than 1 core processor is available. In the case of this study a server was used allowing 7 simultaneous individuals being processed in each turn. In this way the evaluation of a population with size of 20 took 3 turns of the processor. The initial population was generated randomly. Despite it isn't the best approach

Table 4. Optimization sample space.

PCOMP	Number of thicknesses variables	Number of angles variables	t_base	$\theta$ _base
200	14	14	3	3
201	1	1	3	3
202	1	1	3	3
203	1	1	3	3
204	1	1	3	3
205	1	1	3	3
206	1	1	3	3
300	12	12	3	3
301	1	1	3	3
302	1	1	3	3
303	1	1	3	3
304	1	1	3	3
305	1	1	3	3
306	1	1	3	3
400	11	11	3	3
401	1	1	3	3
402	1	1	3	3
403	1	1	3	3
404	1	1	3	3
405	1	1	3	3
406	1	1	3	3
500	11	11	3	3
501	1	1	3	3
502	1	1	3	3
503	1	1	3	3
504	1	1	3	3
505	1	1	3	3
506	1	1	3	3

for increasing optimization efficiency (Castro, 2009), the aim is to compare MOGAII performance in the worst case, i.e. having no idea of the initial laminate, to prove its worthiness not only for optimizing but also for designing and sizing a laminate. The best efficiency is achieved by including in the initial population some promising designs, avoiding the initial search in the sample space taken in first generations (Castro, 2009). The problem of inserting known promising designs is the bias included in the optimization search. This could reduce the robustness of the method considerably if the genetic algorithm parameters aren't set correctly (Castro, 2009).

Table 5. Parameters used in the MOGAII optimization.

Number of Initial Individuals	20	Number of Generations	737
Probability of Directional Cross-Over	0.8	Probability of Conventional Cross-Over	0.0
Probability of Selection	0.1	Elitism	Enabled
Probability of Mutation	0.1	DNA String Mutation Ratio	0.1
Random Seed	1	Concurrent Runs	7

### 3. RESULTS

#### 3.1 modeFRONTIER<sup>®</sup> Results

The weight evolution is showed in Fig. (4 - a). The convergence pattern of this optimization is showed in Fig. (4 - b). The optimization goes fast until approximately 100 generations. This is 14% of the total of 737 generations. In this first stage the convergence has achieved 90%. This can be seen in figures "a" and "b". Beyond this level of convergence further decrease in the structural weight only comes after spending many generations with unchanged minimum weight. This fact reveals which operation acts more in each stage of the optimization. From the pattern showed in Fig. (4 - b) cross-over and selection operators drove the optimization until a level of approximately 90% of the convergence. It can be suggested that mutation operator acted as the main process for further reduction of structural weight. As this operator randomly changes the variables of some individuals around the values coming from the parents, finding a better result for next generations usually takes longer time when comparing to cross-over and selection.

As can be seen in Fig. (4 - a) the heavier laminate of the first generation that was found randomly has a total structural weight of 1,083.93 kg and a buckling load factor of 5.227. The final best result is 462.16 kg and has a buckling load factor of 1.037. This means a weight reduction of 57.36% and 3.7% of margin of safety left.

#### 3.2 GENESIS<sup>®</sup> Results

The finite element optimization analysis performed in GENESIS<sup>®</sup> achieved a soft convergence after 50 design cycles, as showed in Fig. (5). It is showed for each design cycle the values for the objective function and constraint violation. The mass evolution during the optimization doesn't follow a smooth pattern, not even continuous, as shown in Fig. (5). It happens due to the nature of this optimization, which deals with discrete thicknesses and angles, those seeming not to be

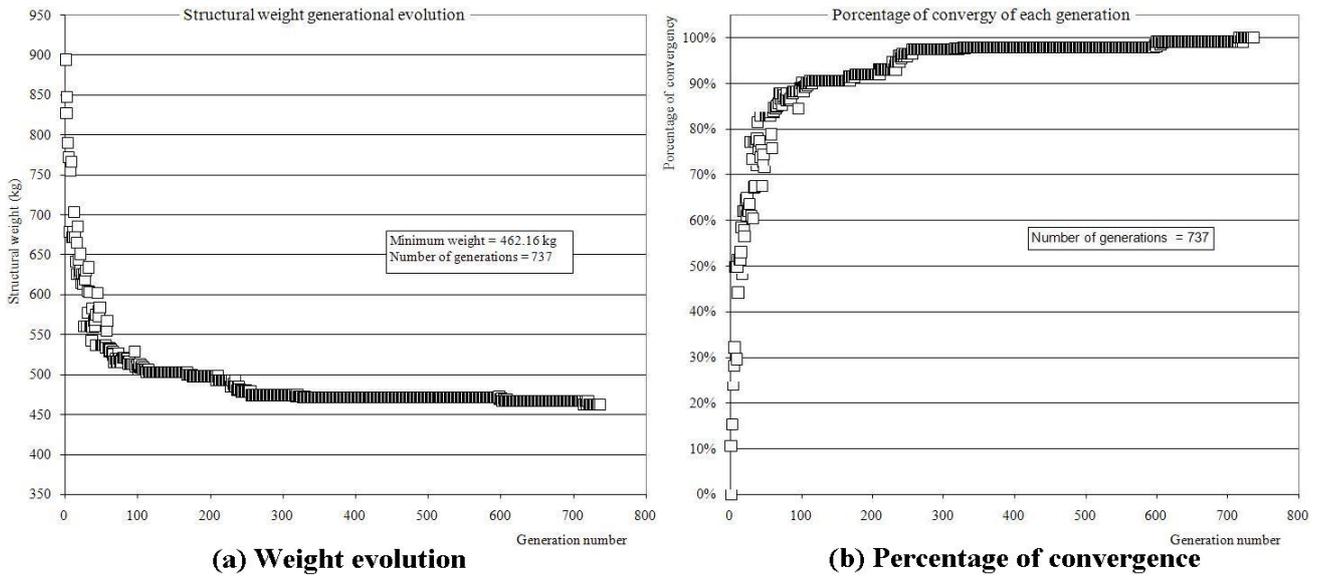


Figure 4. MOGAII optimization results.

weathered by BIGDOT algorithm. The results showed almost 30% of mass reduction by the optimization process.

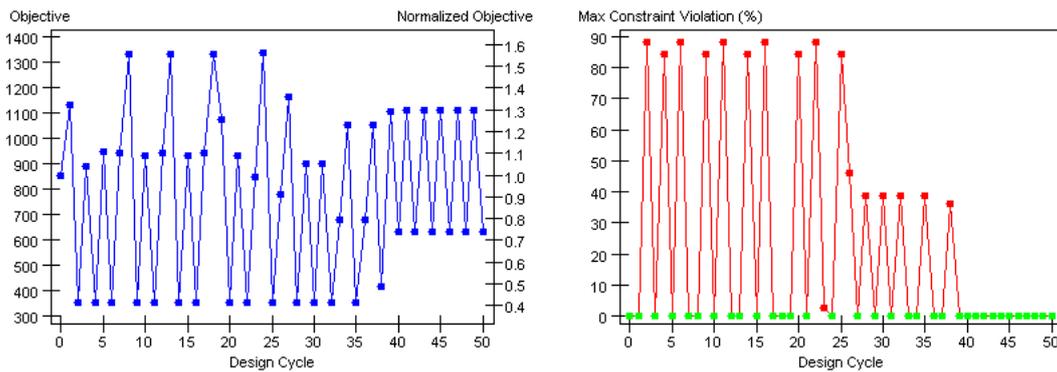


Figure 5. GENESIS results

### 3.3 Comparison of results

MOGAII do not need an initial design while BIGDOT needs a starting laminate in order to perform the optimization process. In the case of applying Guimaraes' methodology (Guimaraes *et al.*, 2010) an initial laminate is already necessary to generate good templates for the laminate entries ("PCOMP" in the case of using NASTRAN or GENESIS' solver), avoiding unnecessary variables.

Both algorithms showed a good weight reduction along the optimization, but MOGAII achieved a more significant percentual reduction. MOGAII heavier desing of first generation weighed 1,083.93 kg and the final best design 462.16 kg 57.36%. The initial laminate given to BIGTOD's optimization weighed 854,64 kg and the final best design 633.48 Kg. The margin of safety left by MOGAII is 3.7% while BIGDOT left 9.8%.

The resulting laminates for both methods are shown in Tab. (6). In a real manufacture a laminate usually has no more than 4 consecutie plies of the same orientation. This practice avoids delamination by increasing transversal bonding among the plies. MOGAII results show a higer variety of angles and a higher shuffling among the four orientations used in the design. On the other hand the BIGDOT's results have lower variety of angles, and consequently a lower shuffling. The results generated by BIGDOT' would need the addition of some plies in different angles. Results from MOGAII seem more readily feasible in the case of a real manufacture.

Table 6. Stacking Sequence Results

Laminate	modeFRONTIER Results	GENESIS Results
200	$[(45_2/-45_2)_2/90_2/(45_1/-45_1)_2/90_2/45_1/-45_1/90_4]_s$	$[90_{20}]_s$
201	$[(45_2/-45_2)_2/90_2/(45_1/-45_1)_2/90_2/45_1/-45_1/90_4]_s$	$[90_{24}]_s$
202	$[(45_2/-45_2)_2/90_2/(45_1/-45_1)_2/90_2/45_1/-45_1/90_4]_s$	$[90_{24}]_s$
203	$[(45_2/-45_2)_2/90_2/(45_1/-45_1)_2/90_2/45_1/-45_1/90_4]_s$	$[90_{24}]_s$
204	$[(45_2/-45_2)_2/90_2/(45_1/-45_1)_2/90_2/45_1/-45_1/90_4]_s$	$[90_{24}]_s$
205	$[(45_2/-45_2)_2/90_2/(45_1/-45_1)_2/90_2/45_1/-45_1/90_4]_s$	$[90_{24}]_s$
206	$[(45_2/-45_2)_2/90_2/(45_1/-45_1)_2/90_2/45_1/-45_1/90_4]_s$	$[90_{24}]_s$
300	$[45_1/-45_1/0_2/90_2/45_1/-45_1]_s$	$[45_6/0_4/-45_6/0_4]_s$
301	$[45_1/-45_1/0_2/90_2/45_1/-45_1]_s$	$[45_6/0_4/-45_6/0_4]_s$
302	$[45_1/-45_1/0_2/90_2/45_1/-45_1]_s$	$[45_6/0_4/-45_6/0_4]_s$
303	$[45_1/-45_1/0_2/90_2/45_1/-45_1]_s$	$[45_6/0_4/-45_6/0_4]_s$
304	$[45_1/-45_1/0_2/90_2/45_1/-45_1]_s$	$[45_6/0_4/-45_6/0_4]_s$
305	$[45_1/-45_1/0_2/90_2/45_1/-45_1]_s$	$[45_6/0_4/-45_6/0_4]_s$
306	$[45_1/-45_1/0_2/90_2/45_1/-45_1]_s$	$[45_6/0_4/-45_6/0_4]_s$
400	$[0_2/90_2/45_2/-45_2/0_2/45_1/-45_1]_s$	$[0_{20}]_s$
401	$[0_2/90_2/45_2/-45_2/0_2/45_1/-45_1]_s$	$[0_{24}]_s$
402	$[0_2/90_2/45_2/-45_2/0_2/45_1/-45_1]_s$	$[0_{24}]_s$
403	$[0_2/90_2/45_2/-45_2/0_2/45_1/-45_1]_s$	$[0_{24}]_s$
404	$[0_2/90_2/45_2/-45_2/0_2/45_1/-45_1]_s$	$[0_{24}]_s$
405	$[0_2/90_2/45_2/-45_2/0_2/45_1/-45_1]_s$	$[0_{24}]_s$
406	$[0_2/90_2/45_2/-45_2/0_2/45_1/-45_1/0_2]_s$	$[0_{24}]_s$
500	$[45_2/-45_2/0_8]_s$	$[0_{20}]_s$
501	$[45_2/-45_2/0_8]_s$	$[0_{20}]_s$
502	$[45_2/-45_2/0_8]_s$	$[0_{20}]_s$
503	$[45_2/-45_2/0_8]_s$	$[0_{20}]_s$
504	$[45_2/-45_2/0_8]_s$	$[0_{20}]_s$
505	$[45_2/-45_2/0_8]_s$	$[0_{20}]_s$
506	$[45_2/-45_2/0_8]_s$	$[0_{20}]_s$

#### 4. CONCLUSIONS AND FURTHER COMMENTS

A method to optimize the stacking sequence and the number of layers for unstiffened anisotropic laminated fiber composite plates was applied in two comercial softwares: modeFRONTIER<sup>®</sup> and GENESIS<sup>®</sup>.

Results from MOGAII seems more reliable and applicable than those of BIGDOT.

The computational cost of BIGDOT is considerable lower then MOGAII's.

The convergence pattern of MOGAII is smooth and clear while BIGDOT ended the optimization lacking convergence.

A good result can be obtained by MOGAII after the first stage of optimization, lowering the computational cost considerably without too much loss. In this study at 14% of the optimization the MOGAII achieved 90% of the convergence, taking most of its turns for smaller gains.

One way to improve the whole optimization is to use each thickness varying only from 0 to 1\*t instead of the adopted variation among 0 or 1\*t or 2\*t. Let's name the former case as case B and the last as case A, that was used in the present study. In case B the number of variables (called "n") increase since the representation of a given laminate, e.g. with 40 plies will need double the amount of variables. The base values for thickness and angle (called "t\_base" and "θ\_base") will decrease, being in case B equivalent of 2/3 of case A. The relations presented in Eq. 3 to Eq. 5 demonstrate how does the new sample space is a function of the current used in this study. Applying Eq. 5 the new sample space, i.e. case B, equals  $(4/9)^{2*72}(5.1*10^{68})^2 = 5.0*10^{86}$ , that is considerably higher. It would probably let to a better laminate, but would surely increase the computational cost of the optimization.

$$SE_A = (t\_base_A * \theta\_base_A)^n \quad (3)$$

$$t\_base_B = \frac{2}{3} * t\_base_A \text{ and } \theta\_base_B = \frac{2}{3} * \theta\_base_A \quad (4)$$

$$SE_B = (t\_base_B * \theta\_base_B)^{2n} = (\frac{2}{3} * t\_base_A * \frac{2}{3} * \theta\_base_A)^{2n} = (\frac{4}{9})^{2n} * (SE_A)^2 \quad (5)$$

## 5. ACKNOWLEDGEMENTS

To EMBRAER for letting GENESIS<sup>®</sup> and modeFRONTIER<sup>®</sup> licenses available.

## 6. REFERENCES

- Bruhn, E.F., 1965. *Analysis ad Design of Fligh Vehicle Structures*. Tri-State Offset Co., Cincinnati, USA.
- Buckles, B.P. and Petry, F.E., 1992. *Genetic Algorithms*. IEEE Computer Society Press, Los Alamitos, USA.
- Castro, S.G.P., 2009. *Otimização Eficiente the Asa em Material Compósito via Algoritmo Genético*. Ph.D. thesis, ITA - Instituto Tecnológico de Aeronáutica, São José dos Campos, Brazil.
- ESTECO, 2003. *modeFRONTIER 3.2.0 User Manual*. ESTECO, Trieste, Italy.
- Goldberg, D.E., 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Publishing Company, Inc., USA.
- Guimaraes, T.A.M., Ravetti, F.M.N. and Hernandez, J.A., 2010. "Evaluation of a simple approach for the optimization of the stacking sequence, thickness and angles of composite laminated structures".
- Haftka, R.T. and Walsh, J.L., 1992. "Stacking sequence optimization for buckling of laminated plates by integer programming". *AIAA Journal*, Vol. 30, No. 3, pp. 814–819.
- Herencia, J.E., Weaver, P.M. and Friswell, M.I., 2007. "Optimization of long anisotropic laminated fiber composite panels with t-shaped stiffeners". *AIAA Journal*, Vol. 45, No. 10, pp. 2497–2509.
- Jones, R.M., 1999. *Mechanics of Composite Materials*. Taylor Francis, Philadelphia, USA.
- MSC, 2008. *MSC NASTRAN 2008r2 - Quick Reference Guide*. MSC, Santa Ana, USA.
- Nagendra, S., Jestin, D., Gürdal, Z., Haftka, R.T. and Watson, L.T., 1996. "Improved genetic algorithm for the design of stiffened composite panels". *Computers and Structures*, Vol. 58, No. 3, pp. 543–555.
- Schimit, L.A. and Farshi, B., 1973. "Optimum laminate design for strength and stiffness". *Internatinal Journal for Numerical Methods in Engineering*, Vol. 7, No. 4, pp. 519–536.
- Soremekun, G.A.E., 1997. *Genetic Algorithms for Composite Laminate Design and Optimization*. Ph.D. thesis, Faculty of the Virginia Polytechnic Institute and State University, Blacksburg, USA.
- Tsai, S.W., 2008. *Theory of Composites Design*. Think Composites, Stanford, USA.
- VRD, 2010. "Bigdot - large scale optimization software library". 25 Mar. 2010 <<http://www.vrand.com/BigDOT.html>>.

## 7. COPYRIGHTS

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