COMPOSITE WING OPTIMIZATION WITH PROGRESSIVE MESH REFINEMENT

Saullo Giovani Pereira Castro, saullo.castro@embraer.com.br
José Antônio Hernandes, hernandes@ita.br
Eliseu Lucena Neto, eliseu@ita.br
Instituto Tecnológico de Aeronáutica
Praça Marechal Eduardo Gome, 50 – Vila das Acácias –CEP 12.228-900-São José dos Campos, SP

Abstract. *The optimized subject of this article is a composite wing where the variables are the number of plies in four different orientations, -45º, +45º, 0º and 90º composing the laminates and the orientation of these plies. The measured output variables are the critical buckling load factor and the structural weight, the first used as design constraint for finding feasible and non feasible designs, the second for finding the best design among feasible ones. It is applied a usual mutation ratio (10% of the population is mutated with a string mutation ratio of 5%) and suggested a methodology for speeding up the convergence of optimization. The refinement is progressively done as the optimization goes on until the final refinement. The computational cost of this progressively refined optimization is compared with the computational cost of the optimization using refined mesh since the beginning. It was also studied the effect of the number of generations used with the coarser mesh in the final best result for the structural weight, and then the importance of the initial search was noticed. It is taken the advantage of fast running – low computational cost – using coarser mesh for searching sample space combined with the reliable results from the refined mesh. As the sample space was better explored the results given by progressively refined optimization are expected to be better than those from the optimization using refined mesh only.*

Keywords: *Genetic Algorithm, composite, optimization, wing*

1. INTRODUCTION

Since nineteen fifties computational scientists have been studying evolutional systems keeping in mind that mechanisms relative to evolution could be used as a tool for optimization in daily engineering problems.

The required processing capability to perform such a task in a practical and cheaper way has been available since 80ties, with a new generation of high speed low cost computers. Since then engineers have applied computer assisted optimization widely by using algorithms that couldn't be used earlier.

Despite the fact computational costs get lower the problems get proportionally more complex. Due to this the issue of efficiency of optimization algorithms has been kept alive. A very effective algorithm, which finds the best solution, but not very efficiently, with high processing time, may be better replaced by a less effective and more efficient one, that would find a good solution faster.

The eingenvalue solution for finding critical buckling load factor in finite element analysis may be computationally very demanding. In these problems the limited computational resources can lead a multivariable analysis to a poor solution far away from the best solution, since the sample space search may not be conveniently explored. This fact clearly limits the widespread usage of optimization in large finite element problems with many design variables.

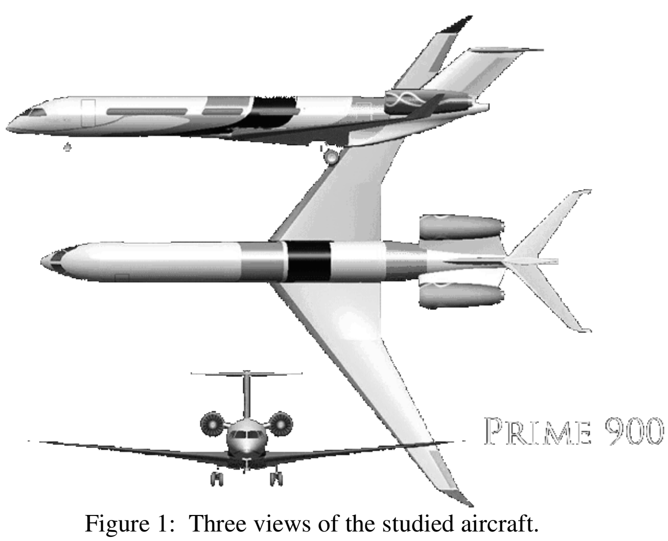


Figure 1: Three views of the studied aircraft.

Castro, 2009 studied the optimization of composite structures through the use of genetic algorithms. The chosen structure is a composite wing of an ultra long range aircraft studied by EMBRAER Program of Engineering Specializing, group 10. Basically the aircraft cruises at Mach 0.92 in a maximum altitude of 51000 ft to reach the maximum range of 7460 nm (nautical miles). The maximum takeoff weight of this airplane is 95000 lb. Figure 1 shows the three views of the airplane. The fuel tank volume is totally contained in the wing structure. Figure 2 shows the structural main components of the wing. The rear spar has a kink at the wing break to increase available fuel tank volume in the wing box and in the wing stub (below fuselage floor).

The optimization objective is basically the minimization of the structural weight. This is performed by finding the best laminates build up the wing. As the wing load increases from the tip to the root it is also expected for the wing root laminates (near fuselage) to be more reinforced. So the optimum laminate for the wing root is not the optimum laminate for the wing tip. Surely for the wing tip it will be thinner and lighter when compared to the root laminates. A good optimization need to take this fact into account to allow further weight reduction. To allow further optimization the laminates for the rear spar, front spar, upper skin and lower skin were divided in three zones, called zone 1, zone 2 and zone 3. Landing gear extra spar also has a laminate, but due to the smaller size of this spar the laminate was optimized considering just one zone. Figure 4 shows these zones.

For each zone the variables are the number of plies for each of the four chosen orientations: 0º, 90º, -45º and +45º; relatively to the laminate main direction. This main direction also is a variable for the upper and lower skins. Figure 4 shows the main direction of the laminate (X1). For the spars the laminate main direction is chosen as the longitudinal direction of the respective beam (parallel to the neutral axis). The zones closer to the tip will be the base laminate for the zones closer to the root. For these zones necessary plies are added as reinforcement for the less loaded zones. Figure 5 shows the adopted stacking sequence scheme, which allows the chosen laminates to be manufactured through automated processes, such as automatic tape laying (ATL).

Table **1** shows the total number of laminate variables for all the wing components. The possible plies orientations are: 0º, 90º, -45º and +45º; so there are 4 variables. The optimization problem involves 54 variables.

Table 1. Laminate Variables for optimization.

	Lower Skin	Upper Skin	Frontal Spar	Rear Spar	Landing Gear Spar
Laminate direction variables	1	1	0	0	0
Number of possible plies orientations x number of zones	4x3 = 12	4x3 = 12	4x3 = 12	4x3 = 12	4x1 = 4
Variables per component	13	13	12	12	4
TOTAL variables	**54**				

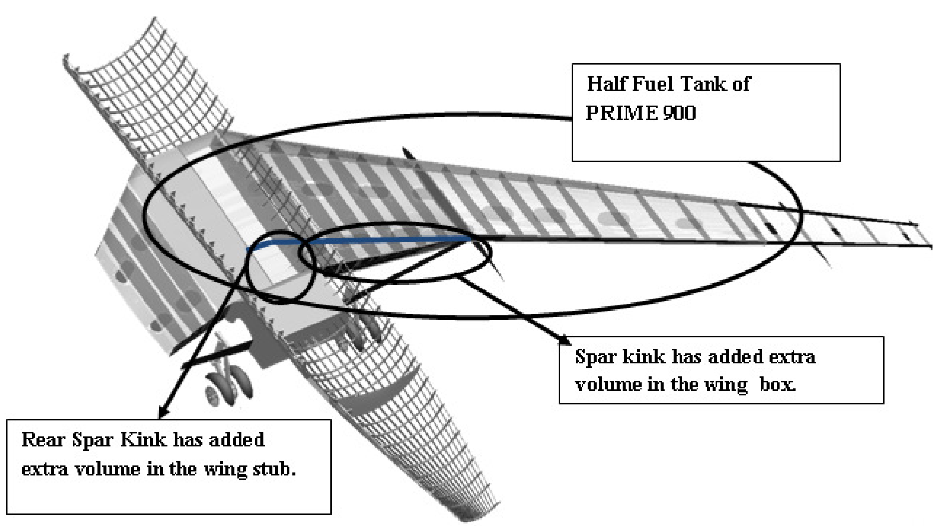


Figure 2: Structure scratch showing a complete view of the wing contained fuel tank.

In the presented wing and laminates it will be tested the suggested methodology studied by Castro, 2009, for optimizing the weight of a composite wing. In his work the number of variables is closer to 300 and it also considers geometry variations (ribs angles and positioning along wing span). This methodology consists in the use of progressive mesh refinement along the optimization. It will be compared with the traditional optimization method using fixed mesh refinement.

The minimum population size here used is a result of the study (Castro, 2009). Figure 3 shows a set of experiments in which the computational cost has been kept constant, varying just the population size and the number of generations inversely. As can be seen the minimum population size which generated stable results was the one with 25 individuals.

Using this minimum population size for the same problem (Castro, 2009) with the reduced number of variables it was possible to eliminate instabilities in convergence. The 10 x 2000 graphic of Figure 3 illustrates one case with instabilities in the results due to insufficient individuals in the population.

In the present article the effect of the number of generation used to explore the sample space with the coarser mesh is also investigated.

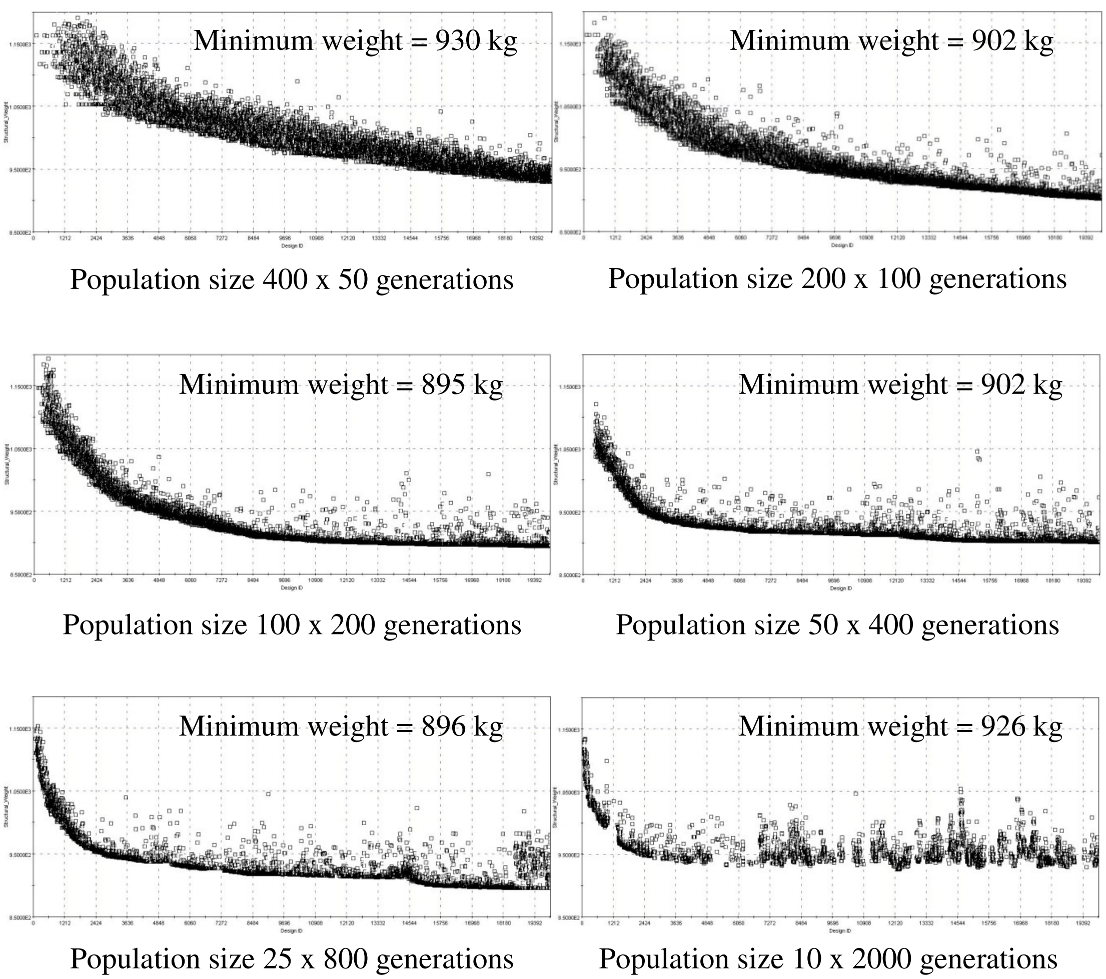


Figure 3: Population size X number of generations with constant computational cost. Copied from Castro, 2009.

The genetic algorithm applied for optimization is a MOGA II (Multi Objective Genetic Algorithm) available in the ESTECO software MODEFRONTIER, version 3.2. This software integrates all analysis steps: mesh generation, solver execution and buckling load factor output and weight output measuring. In the mesh generation step the laminate variable values are inputted for the generation of composite properties for the solver. The solver used here is the Optstructure, available in the Altair Hyperworks software package, release 8.0. This solver is fully compatible with NASTRAN. The composite property card is also the same, PCOMP. So, the task of MODEFRONTIER is only to set adequate PCOMP cards as the optimization runs to find a minimum weight that keep structure constrained with a buckling load factor higher than 1 (one). Each individual being analyzed by the genetic algorithm is consisted of a group of these PCOMP. The chromosome is then consisted of a group of variables which carry the necessary information for stacking all laminates. In other words, all PCOMPs of an individual form the respective chromosome. The best individuals identified along optimization will be preferentially crossed to generate new generations with progressively better laminates, or PCOMP cards.

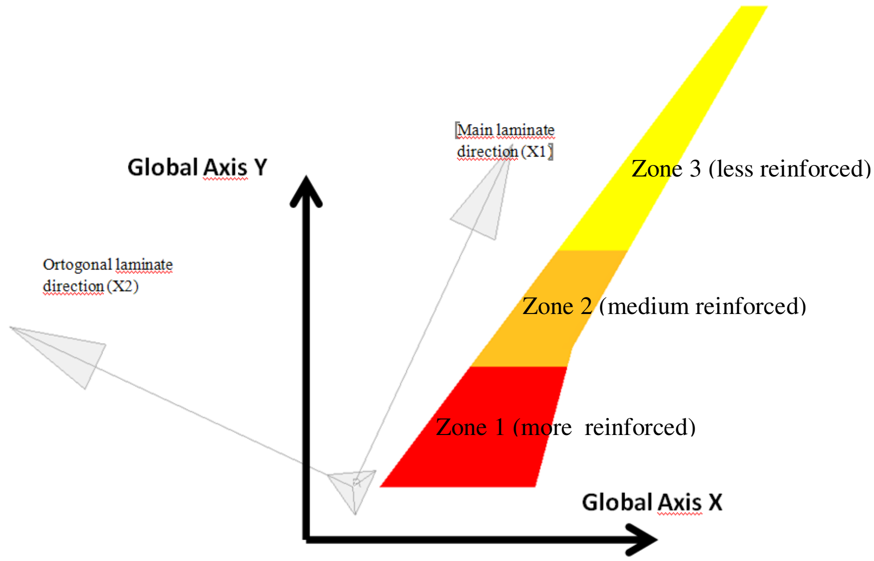


Figure 4: Zones distribution of the laminates.

2. METHODOLOGY

Table 2 shows the set of experiments that has been built in this study. This table shows the ID range for each mesh refinement for each design. It can be noticed that the 'Refined Mesh' design only evaluates runs at the maximum level of mesh refinement used in this article. Designs labeled as 'Progressive' use the same number of runs at mesh refinement levels 2 and 3 and vary the number of runs with the coarsest mesh to evaluate how the step using the coarser mesh, called here as 'sample space searching', affects the result. The last refinement level ('Refinement 4') is used until convergence of the structural weight output variable.

Figure **6** brings the mesh refinement visualization and further information about each refinement level. It can be seen a considerable increase in the average processing time for each run as the mesh gets refined.

All the runs were done in a Personal Computer with the following specification: 2Gb of 1066MHz RAM Memory, 2.7GHz Core to Duo Processor (Intel), Hard Drive of 7000rpm speed.

3.RESULTS

Figure 7 shows the structural weight history chart for each experiment. As the initial population is random, for all experiments this structural weight tends to raise in a first step, where the genetic algorithms is looking for a 'feasible' wing, or a design constrained wing, which has the buckling load factor for all components higher than 1. The weight keeps rising until a feasible design is found. The Fuzzy logical applied for comparing how much a design is far from design constraints has allowed genetic algorithms to go fast into a set of feasible designs. After finding these feasible designs the genetic algorithm then starts looking for the best designs by performing directional crossover, selection and mutation operations. This step goes until the convergence is reached. For all designs evaluated in this study the runs were kept until convergence.

Figure 8 shows the total elapsed time for the experiments shown in Figure 7. As it can be seen, the 'Refined Mesh' experiment which has started and gone until convergence with the refined mesh has an elapsed time one order superior. The remaining experiments have more similar elapsed times. By this great difference among elapsed times it can be said for sure that optimization using refined mesh since the beginning for multivariable problems isn't the best way to go. The penalty by going this way is the computational cost increase.

Some experiments were allowed to run long time after convergence. Due to this the direct comparison of the total elapsed time is not correct. It would be correct only if a convergence criterion was applied for stopping evaluated designs. This criterion wasn't developed and it isn't necessary since the comparison can be done by finding the run ID in which convergence of structural weight has occurred for each experiment, even considering that the runs can go beyond this point. With this run ID it is possible to calculate the time that would be enough for reaching convergence multiplying the average time by the number of runs. Table 3 shows these values of 'time that would be enough' to all experiments.

The computational cost is then calculated as the time that would be enough by the run ID at convergence, giving the units of ‘seconds’. If the user wants to convert this value to monetary units it is only necessary to know the price of keeping a computed running a second, giving unit of $$/second, multiplying this by the computational cost in seconds.

The experiment using only refined mesh has shown a high computational cost and it is curious the fact it hasn’t shown the best weight result among the six experiments. Surprisingly, it has shown the worst result, with the highest final weight of 1011 kg and the highest computational cost of 97.5 seconds.

Among experiments with progressive refined mesh the more experiments analyzed in the Refinement 1 level the better the results for the minimum structural weight. These better results come with a very small penalty in the computational cost. For each 500 runs evaluated in the Refinement 1 level the optimization time is increased about one hour in the CPU used. The best experiment is ‘Progressive 5’ which reached the minimum weight of 950 kg with a final buckling load factor of 1.0036. The computational cost of this experiment was 9.5 seconds, 10 times lower than experiment using only the refined mesh.

In a real optimization the user would face the doubt of keeping raising the number of generations until full convergence is reached or stopping earlier due to computational costs limitations as done here. The best approach is to optimize until the global minimum is found for the current mesh refinement level, but costs limitations would probably limit this approach. A control algorithm can be coded to monitor the convergence for each refinement level. An adequate convergence criteria such as weight variation below 1 kg, for example, will then allow further refinement to be started.

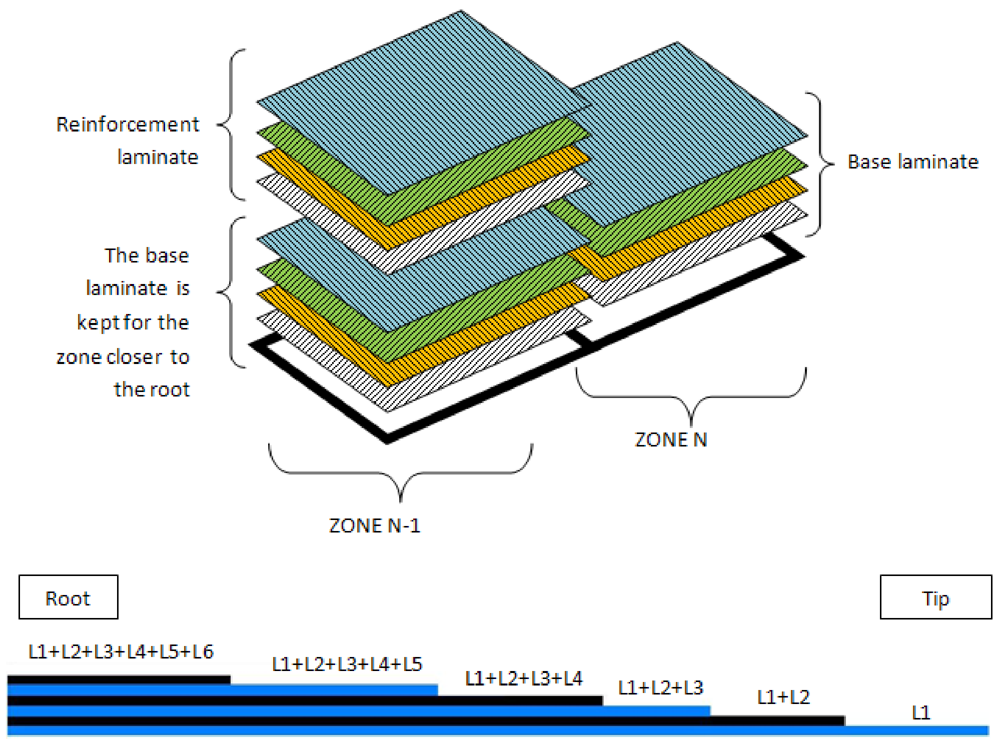


Figure 5: Stacking sequence for the laminate, showing consistence with automated lamination processes.

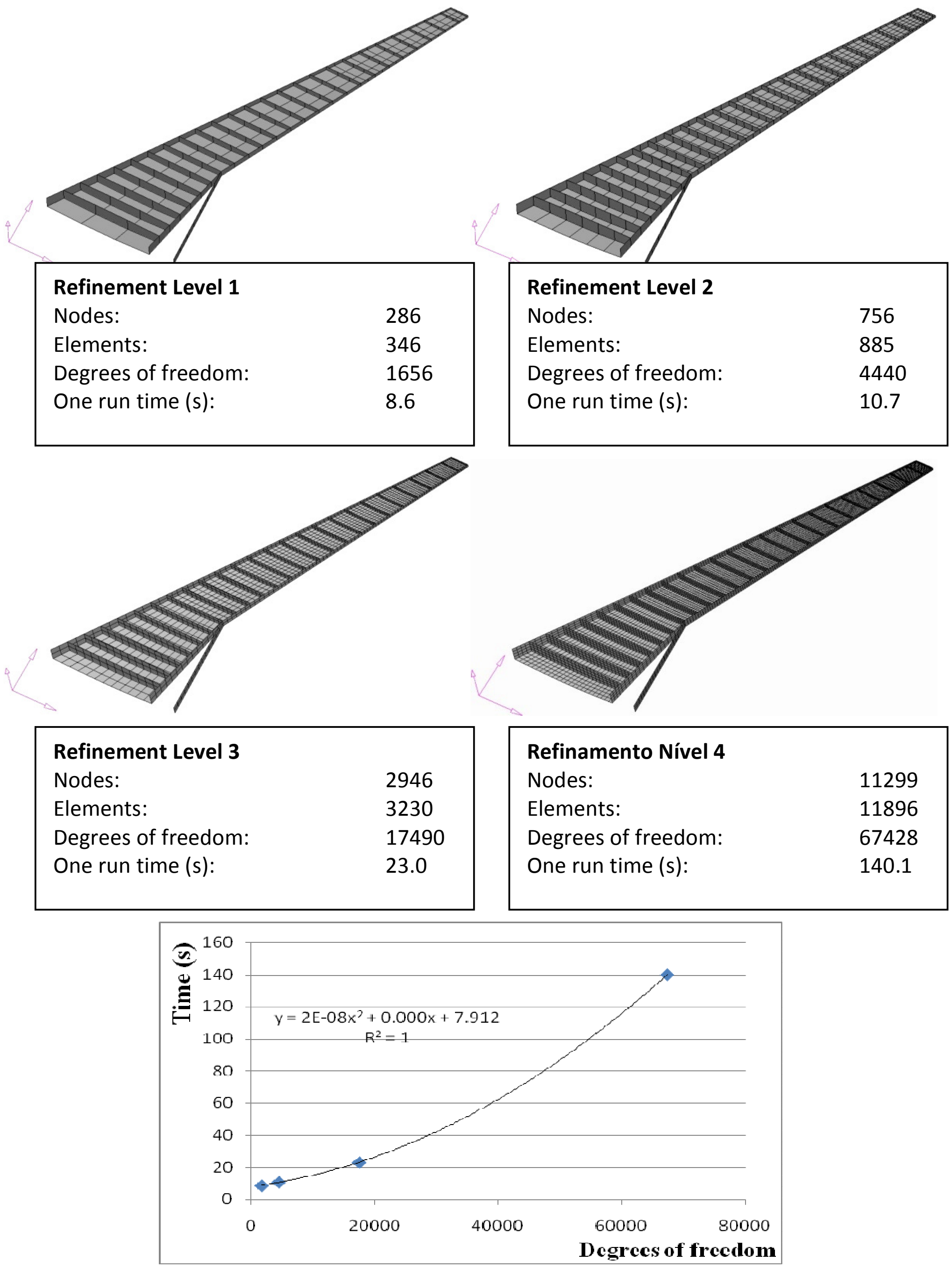
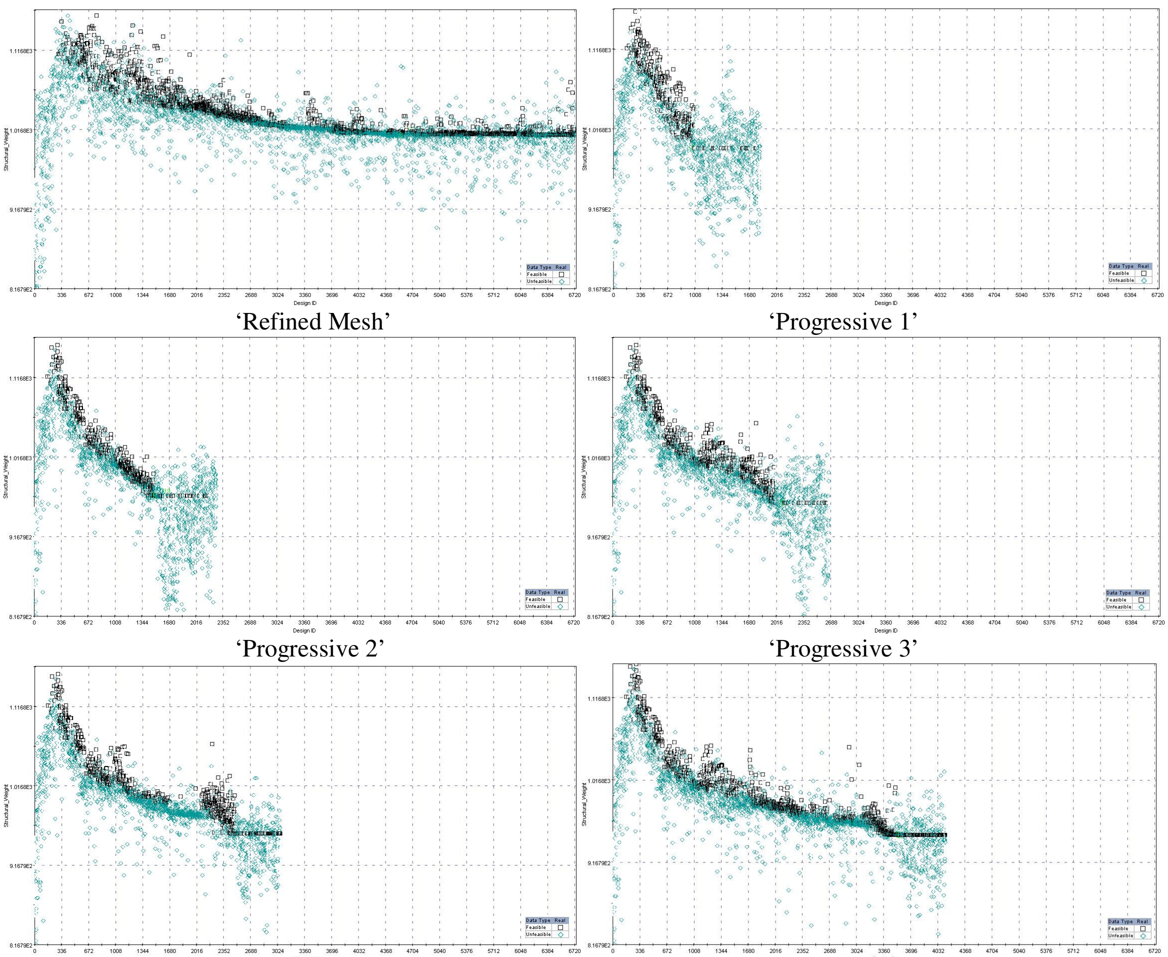


Figure 6: The four mesh refinements used in this article.

Table 2. Experiment set showing that number of runs with mesh at "Refinement 1" level is increased from 0 in "Refined Mesh" design to 3000 in "Progressive 5" design.

		Refined Mesh	Progressive 1	Progressive 2	Progressive 3	Progressive 4	Progressive 5
Refinement 1	ID min	0	0	0	0	0	0
	ID max	0	499	999	1499	1999	2999
Refinement 2	ID min	0	500	1000	1500	2000	3000
	ID max	0	999	1499	1999	2499	3499
Refinement 3	ID min	0	1000	1500	2000	2500	3500
	ID max	0	1499	1999	2499	2999	3999
Refinement 4	ID min	0	1500	2000	2500	3000	4000
	ID at end	∞	∞	∞	∞	∞	∞

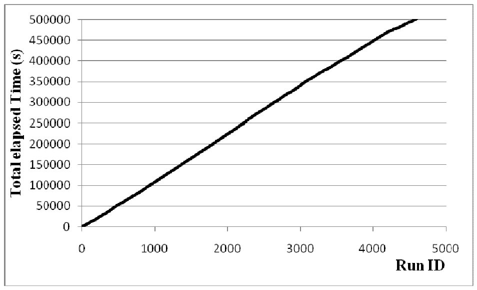


'Refined Mesh' 'Progressive 1'

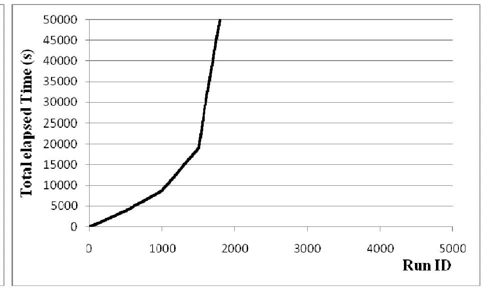
'Progressive 2' 'Progressive 3'

'Progressive 4' 'Progressive 5'

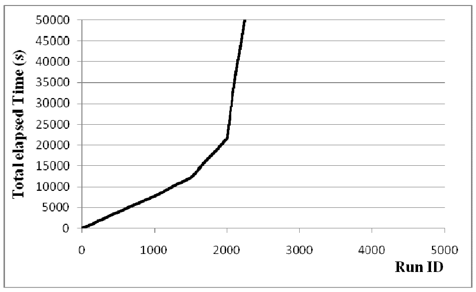
Figure 7: Population size X number of generations with constant computational cost.



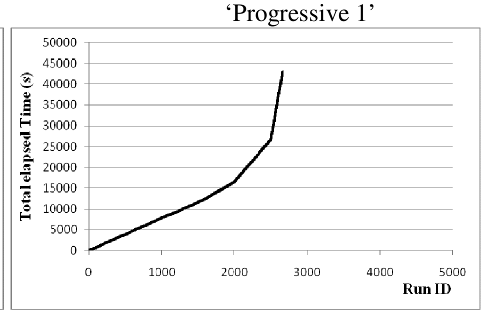
‘Refined Mesh’



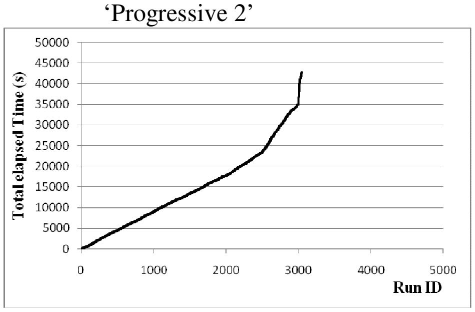
‘Progressive 1’



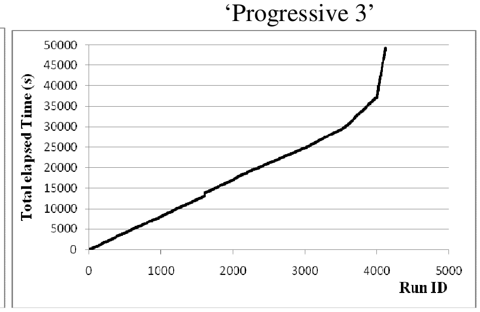
‘Progressive 2’



‘Progressive 3’



‘Progressive 4’



‘Progressive 5’

Figure 8: Elapsed time for the six designs evaluated. Copied from Castro (Reference 0)

Table 3 Resume of the results.

		Refined Mesh	Progressive 1	Progressive 2	Progressive 3	Progressive 4	Progressive 5
Refinement 1	ID min	0	0	0	0	0	0
	ID max	0	499	999	1499	1999	2999
	Average Time/ID (s)	0.0	8.0	8.1	8.3	9.6	9.0
	Minimum mass at Level 1 (k	0	1065	1012	1001	996	965
Refinement 2	ID min	0	500	1000	1500	2000	3000
	ID max	0	999	1499	1999	2499	3499
	Average Time/ID (s)	0.0	10.4	9.9	10.5	12.0	10.8
	Minimum mass at Level 2 (k	0	993	968	960	957	950
Refinement 3	ID min	0	1000	1500	2000	2500	3500
	ID max	0	1499	1999	2499	2999	3999
	Average Time/ID (s)	0.0	22.0	20.3	21.8	29.0	22.1
	Minimum mass at Level 3 (k	0	993	968	960	957	950
Refinement 4	ID min	0	1500	2000	2500	3000	4000
	ID at end	6735	1807	2266	2659	3052	4121
	Average Time/ID (s)	127.5	112.6	120.3	112.0	225.6	130.1
	Mass at convergency	1011	993	968	960	957	950
	Total Elapsed Time	656534	51354	57734	43134	42804	49397
	ID at convergency	6111	999	1519	2106	2497	3568
	Time that would be enough	595706	9244	13515	20031	25095	33986
	BLF at convergency	1.0005	1.0004	1.0008	1.0014	1.003	1.0036
	Computational cost	97.5	9.2	8.9	9.5	10.0	9.5

4. CONCLUSION

The idea of using progressive mesh refinement was applied to the wing structural optimization with the hope of getting faster optimization, therefore the observed gain in weight was totally unexpected when comparing the progressive mesh refinement experiments to the experiments with the fixed most refined mesh. It is believed that the better results for structural weight when changing from constant mesh to progressive refinement are due to the extra sample space searching that occurs when jumping from the current to the next mesh refinement level. In the jump the current designs become unfeasible. In that way some generations have to be run until results become feasible again. The unfeasible initial designs can be explained by the fact that a more refined mesh lead to a less rigid structure, needing more composite laminated plies. It was also observed that among all experiments using progressive refinement the better results were given by the ones with populations with more individuals in the first level of mesh refinement, as expected since sample space exploration is higher for these experiments.

The reduction of the computational cost by 10 times when using progressive mesh refinement supports the application of this methodology for complex designs involving a high number of variables.

5. AKNOWLEDGEMENTS

The authors are grateful to Fundação Casimiro Montenegro Filho and to Embraer for the support during the master's of the first author and to Capes.

6. REFERENCES

Castro, S. G. P., Hernandes, J. A, 2009, "Otimização Eficiente de Asa em Material Compósito Via Algoritmo Genético", São José dos Campos: ITA. 192 p. Dissertação (Mestrado Profissionalizante de Engenharia Aeronáutica e Mecânica) – Instituto Tecnológico da Aeronáutica, São José dos Campos.

Turma 10 do Programa de Especialização em Engenharia (PEE), 2008. EMBRAER. PDR – Preliminary Design Review. São José dos Campos.

Turma 10 do Programa de Especialização em Engenharia (PEE), 2007. EMBRAER. ACR –Aircraft Configuration Review. São José dos Campos.

Neal, J. Y, Blouin, V. Y., Fadel, G. M, 2002, "GA-BASED Multi-Material Structural Optimization Using Stepwise Mesh Refinement". AIAA 2002-5461. 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Atlanta, Georgia.

ModeFrontier version 3.2.0, 2003, "User Manual". ESTECO, Trieste, Italy.

Zuben, F. J. & Attux, R. R. F, 2008, "Computação Evolutiva. Material Complementar da disciplina IA013 – Introdução à Computação Natural", Universidade de Campinas – UNICAMP. <ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/ia013_1s07/topico3comp_07.pdf>.

Poli, R., Langdon, W. B., McPhee, N. F., 2008, "A Field Guide to Genetic Programming". University of Essex, United Kingdom.