

# A hybrid genetic algorithm with local search approach for composite structures optimization

*Leila GHARSALLI\* and Yannick GUERIN\**

*\*DLA/SDT/STM, 52 Rue Jacques Hillairet, 75012 Paris*

*Leila.gharsalli@cnes.fr*

## Abstract

Optimization of a sandwich composite inter stage skirt under mechanical and manufacturing constraints using a hybrid genetic algorithm is studied in this paper in order to improve the performance and the computing time of the classical genetic algorithm. For this purpose, firstly, total number of a skin plies and their orientations angles are considered as design variables, then some complexity is added to enlarge to research space by varying the core thickness of each individual. Two hybridization schemes are proposed where the idea is to treat population in two different ways and evaluate the contribution of each method. Results obtained from this study reveal the fact that the adopted approach based on a hybrid genetic algorithm is highly capable of performing such optimization.

## 1. Introduction

It is well known that material selection plays an important role in engineering design. For example, in aerospace engineering, composite materials, usually defined as the association of two or more materials with different properties to form a useful third material, are considered to be more advantageous than metallic materials owing to their lower densities as well as their high specific stiffness and strength which leads to a mass and cost saving [1][2].

Nevertheless, although such association is more efficient, the raw materials are often expensive. Therefore, the composite structures should be optimized to make the best use of materials. Note that in the mechanical sizing of these structures, plies thicknesses are often predetermined and plies orientations are usually restricted to a small set of angles due to manufacturing limitations. This leads mathematically to a discrete optimization problem that has attracted the attention of many researchers where many solving methods have been proposed [3][4][5][6].

Metaheuristics belonging to the wider family of stochastic methods have been particularly employed to solve this problem [7]. In fact, they are able to deal with discrete variables with a high capacity to find the problem's overall optimum. Besides, unlike most analytical methods, they do not require either a starting point or knowing the gradient of the objective function to achieve the optimal solution. But unfortunately they take relatively long time to find the optimum in those regions which can make their use nonviable depending on the computational cost of each evaluation.

To overcome this limitation, different alternatives have been proposed such as hybrid methods. One of the most popular hybridization techniques is the use of stochastic neighborhood-based local search methods with population-based methods [8][9][10][11] etc.,. In fact, latters are robust and powerful global optimization techniques for solving large-scale problems. On the other hand, local search algorithms are capable to find the local optima with high accuracy and fast convergence but suffers from the problem of foot hills. Hence, a perfect blending between both techniques should offer their advantages while offsetting their disadvantages.

In this perspective, in this work, an association between a well-known based population method: a Genetic Algorithm (GA) [12] and a local search technique resulting in a Hybrid Genetic Algorithm (H-GA) is used to pursue sandwich composite inter stage skirt optimization. Note that GAs, among metaheuristics, have been the most used in the field of composite materials [13][14][15][16]. As hybridization can be done in different manners, two hybridization's schemes are tested herein: firstly, an iterative local search approach is coupled with the GA in a sequential manner. Secondly, GA combines a local search with its own genetic operators in order to reach the optimum.

The paper is organized as follows. The first section is about the formulation of the optimization problem. In the second section, the basics of the genetic algorithm, the chromosome representation and the optimization problem's constraint

handling are described. The proposed hybrid genetic algorithm is presented in the fourth section. The numerical results are shown in the fifth section. Finally, the last section reports the main conclusions that were drawn.

## 2. Optimization problem

We consider the following optimization problem: a sandwich composite inter-stage skirt is located on the upper part of the launcher and is subject to pressure oscillations from engines upstream. In order to limit the vibratory levels at the launcher's top, the skirt has a filtering role. In order to better filtering these oscillations, this structure must have a mechanical sizing according to a compromise between stiffness and flexibility related to the need to avoid buckling. Hence, the optimization's aim is to minimize the total mass while respecting a longitudinal stiffness constraint and buckling strength in addition to composite manufacturing constraints.

Firstly, for the sake of simplicity, sandwich skin plies and core thicknesses are considered as fixed (0.1 mm per ply and 22 mm respectively). In this case, design variables are one skin plies orientations. Afterwards, some complexity is added to the problem by varying the core thickness but keeping the same thickness for plies because it is already chosen based on manufacturable concepts. In this case, design variables are one skin plies orientations plus the core thickness. Note that without any available analytical method to accurately calculate the buckling of the structure, the finite element analysis (FEA) program NASTRAN is used [17].

In the following; the basic GA and two hybrid genetic approaches are performed to solve this optimization problem.

## 3. Genetic algorithm

### 3.1 Basic concepts and principles

GAs, inspired by Darwin's theory of natural selection, have specific mechanics and generally involve genetic operators that intended to improve an initial random population [18]. Note that the population is the set of individuals where an individual is represented by one or more chromosomes. One chromosome is composed of a list of genes containing the hereditary characteristics of the individual and defining its domain. Thus, dealing with a GA means dealing with all this specific vocabulary besides the specific operators such as selection, crossover and mutation.

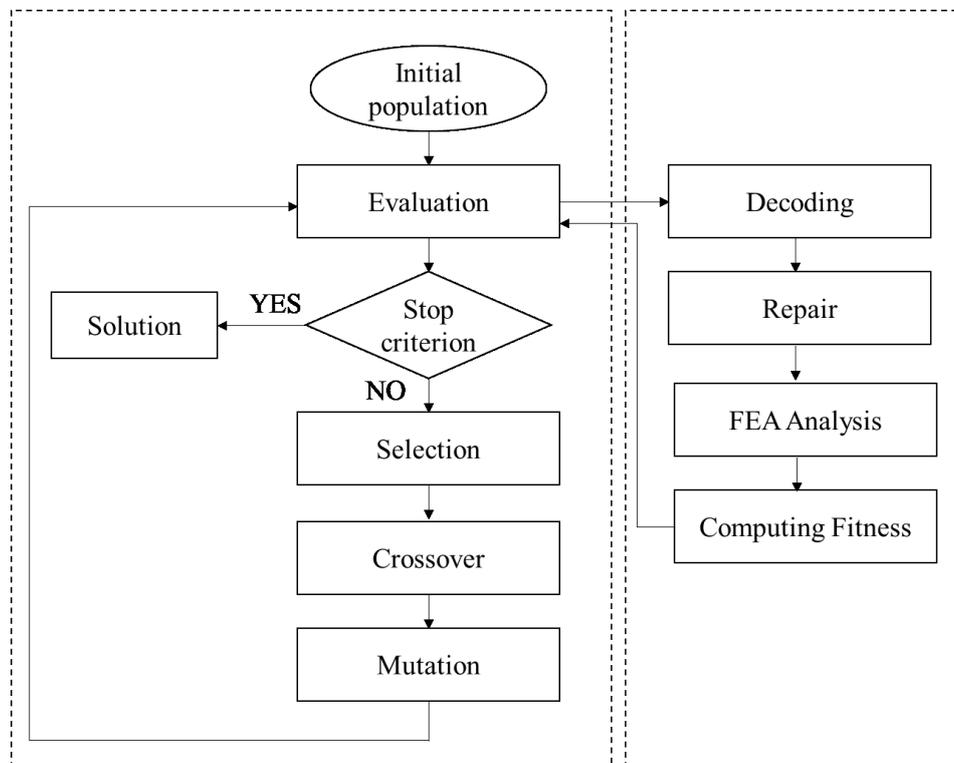


Figure 1 : GA flowchart

Firstly, selection involves a “fitness” function, also known as the evaluation function, that quantitatively measure how fit a given solution is in solving the problem. Many selection methods may be found in the literature but in this work tournament selection is chosen [12]. Then, crossover is the basic genetic operator that involves combining the information from two selected parents to create one or two new individuals. There are many types of crossover operators such as single point, two-point, multi-point, uniform and variable crossover [19][20]. In this work, two-point crossover is applied. Finally, the mutation operator guarantees gene diversity so that the algorithm does not get stuck in local minima [12]. There are also several types of mutation. Herein, the mutation is used either to invert the position of a parent’s genes to create the offspring or to replace the chosen gene value with a randomly selected value in the chromosome domain.

At this point, GA may repeat the process until a stopping condition, which is described later, is satisfied (Figure 1) or, before that, a local search may be pursued, which is the case of this paper. Two local search approaches are employed: firstly, an iterative local search technique is coupled with the GA in a sequential manner. Secondly, GA combines a local search with its own genetic operators in order to reach the optimum.

### 3.2 Chromosome representation

The sandwich composite is obtained from two skins bonded to a core made of a very light material [1]. It is preferred to the monolithic one because it allows to discard the skins that improves the buckling behavior while limiting the longitudinal stiffness. Herein, the initial population is composed of a set of skin plies characterized by orientation angles. Thus an individual in the population is equivalent to a laminate (set of plies). Note that since GA works by coding the problem’s parameters, it can evaluate the ‘fitness’ of every individual and computes the problem’s objectives and constraints based only on that coding without any further knowledge of real parameters. An individual is so represented by one orientation chromosome ( $0^\circ, 90^\circ, \pm 45^\circ, \pm 30^\circ$ ), whose domain contains 5 possible values  $\{0,1,2,3,4\}$  (see Table 1). A second representation of this chromosome is adopted when the core thickness evolving by steps of  $1\text{ mm}$  is part of the design variable. Indeed, in this case, the chromosome is no longer representing only the orientations but its last gene is used to encode the core thickness (see Table 2). In this case, decoding and repairing steps stop at the penultimate gene. Besides, specific genetic operators for continuous variables (simulated binary crossover and uniform mutation) are applied to the last gene.

Table 1: Orientations’ chromosome representation

Encoded	0	1	2	3	4
Decoded	$0^\circ$	$\pm 45^\circ$	$90^\circ$	Empty	$\pm 30^\circ$

Table 2: Second orientations’ chromosome representation

Encoded	0	1	2	3	4	0.22
Decoded	$0^\circ$	$\pm 45^\circ$	$90^\circ$	Empty	$\pm 30^\circ$	0.22

### 3.4 Constraints handling

In GAs, the most common ways of handling constraints are data structure, repair strategies and penalty functions [21]. In this work, the mechanical constraints (buckling, stiffness) are managed during the selection process through tournaments whereas the composite design constraints (mirror symmetry, balancing, grouping etc.,[22]) are overcome by using a repair technique during the orientations’ chromosome decoding step (Figure 2).

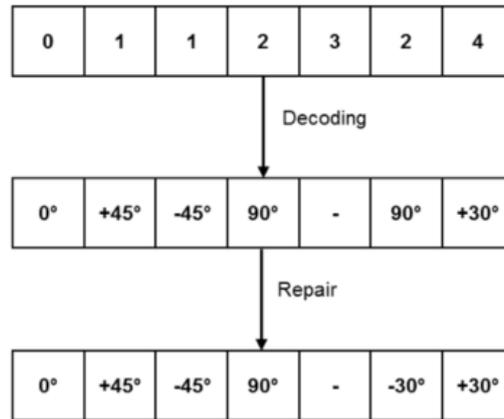


Figure 2: Example of the repair technique

## 4. Hybrid Genetic Algorithms

### 4.1 Local search

Hybrid genetic algorithms (H-GAs) are based on the complementary view of GAs and local search methods. So there are several ways in which a local search can be incorporated in GAs [10][23]. In this work, the first way resulting in the first hybrid algorithm (H-GA-1) consists in a strong cooperation between GA and an iterative local descent search technique for the entire course of the algorithm. The local search is integrated at the beginning of the GA in order to intensify research only around promising solutions because applying the local search to every individual will waste the resources. Figure 3 illustrates the progress of this algorithm where for every new generation, the best individual reached by the GA is selected as an initialization for the local search approach. Then, a set of two neighbors is built following a well-defined structure. Each neighbor is evaluated and if one shows an improved fitness, then it is introduced in the population to replace the first individual and so on.

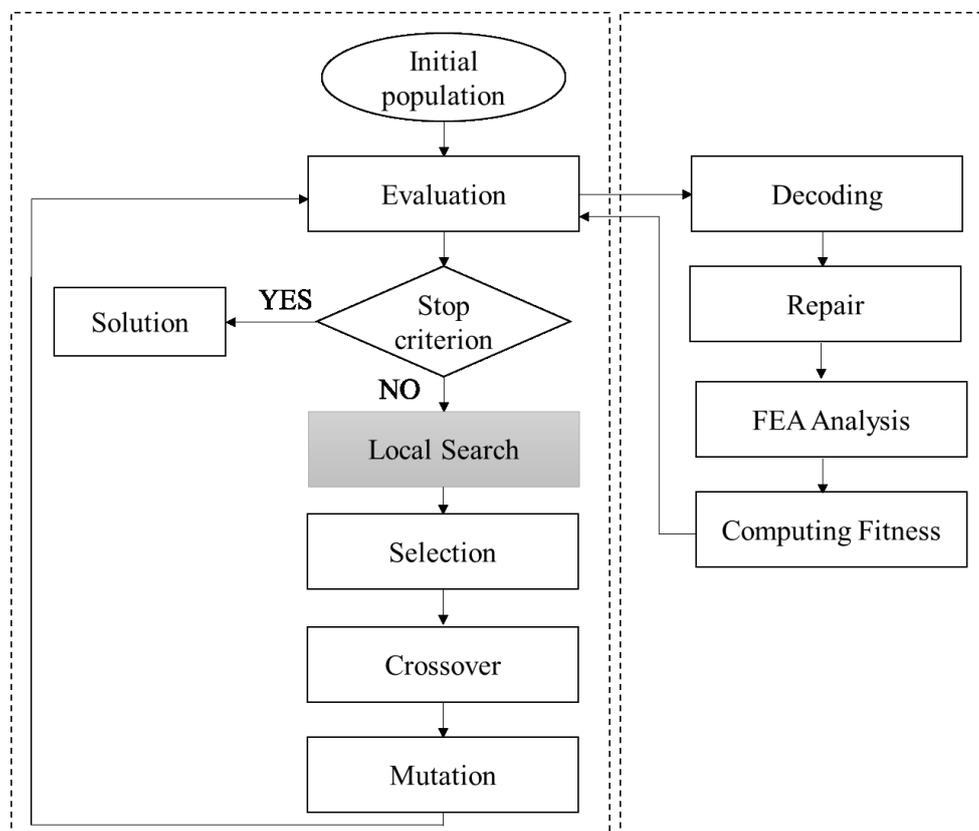


Figure 3: H-GA-1 flowchart

The second way of hybridization resulting in the second hybrid algorithm (H-GA-2) proceeds differently by integrating the exploration capability of the GA and the local search simultaneously. Based on these principle, the idea is to split the population into two equal parts where the first part is explored by the GA and the other part by the local search (Figure 4). An alternate neighborhood is used to peruse the local search based on the fact that a local optimum of one neighborhood structure is not necessary the same for another neighborhood structure and overall optimum is the local optimum of all neighborhood structures. The fitness calculation and genetic operations remain as described before. A variety of stopping criteria can be used to terminate the running of the local search, such as maximum allowed elapsed time, maximum number of iterations, etc. In this paper, local search runs for a fixed number of iterations.

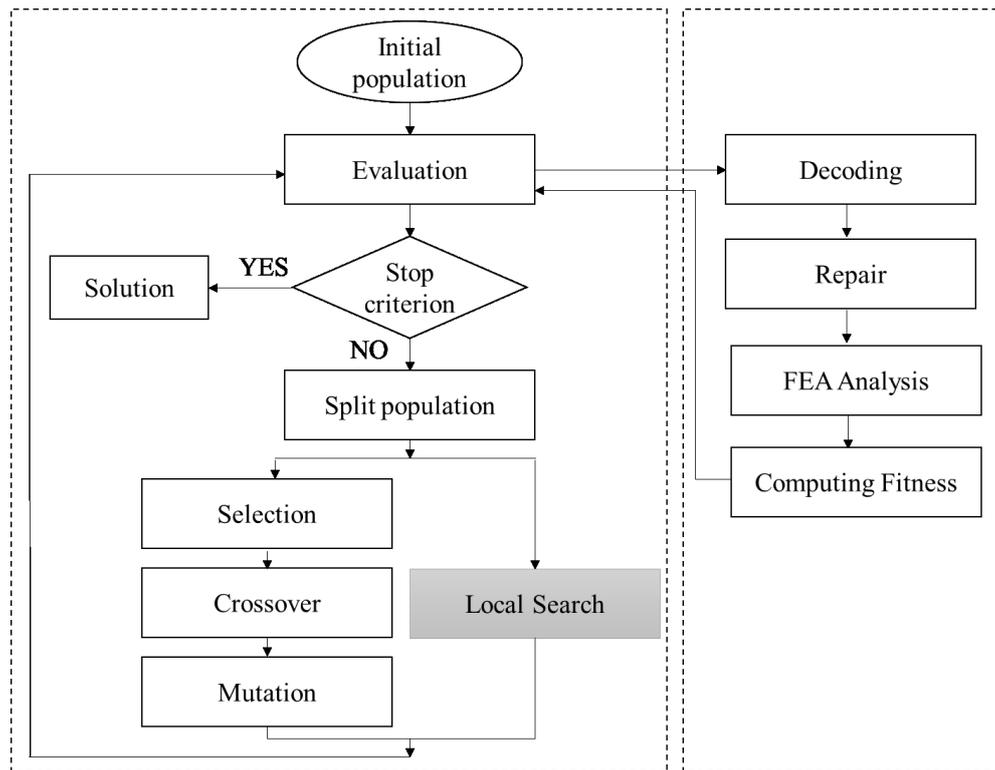


Figure 4: H-GA-2 flowchart

## 4.2 Neighbourhood structures

The neighborhood can be defined in different ways and depends on the type of design variable. As discussed earlier, one individual is represented by one chromosome related to the orientations angles. In this work, two sets of neighbors are proposed: the first neighborhood, used in H-GA-1 and H-GA-2, is created such as for a given individual, a randomly selected gene is flipped to either the previous or the next value in the chromosome's domain. However, if the randomly selected gene is the domain's upper bound (herein 4), then two neighbors are built by flipping firstly the gene to the previous value in the domain (3) then to the lower bound (0) and conversely if the randomly selected gene is the domain's lower bound. Note that these changes are made to verify if there is any neighboring orientation that allows an improvement of the individual's performance. This method is intended to facilitate the fine search in the population against the basic GA which can only use of the mutation operator. Figure 5 shows an illustrative example of this first structure.

Then, for varying the local search and creating more significant disruption to the neighborhood and avoid getting stuck on local optimums, the second neighborhood structure, used in H-GA-2 in alternation with the first neighborhood, is built differently. Indeed, for a given individual, two numbers  $\alpha$  and  $\beta$  are randomly chosen between 1 and the total number of the chromosome genes. if  $\alpha < \beta$ , then the gene with position  $\alpha$  shifts to the right of the gene of position  $\beta$  that remains unchanged in the neighbor. Otherwise, the gene of position  $\alpha$  will be placed before the gene of position  $\beta$  in the chromosome. In both cases, the second neighbor is built as the opposite of the first one. Figure 6 illustrates an example of this construction.

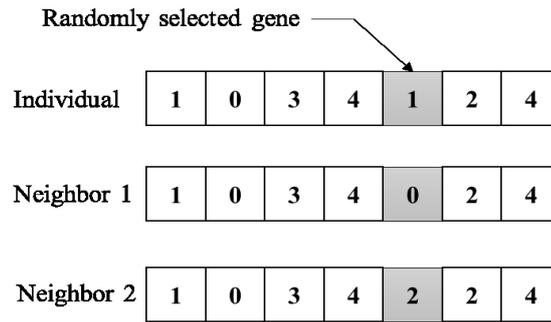
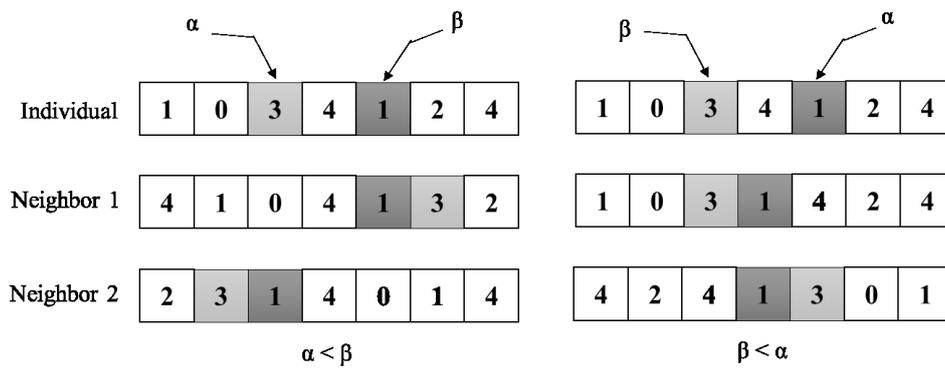
Figure 5: First neighborhood structure (the chromosome' domain is  $\{0,1,2,3,4\}$ )

Figure 6: Second neighborhood structure

## 5. Numerical results

The comparative evaluation of the proposed hybrid algorithms and the classical GA is presented in the following. Note that for the purpose of accelerating the optimization process, a database containing individuals already assessed and their respective fitness is created during the optimization process. In this way, before each evaluation, algorithms run through the database and look for fitness from the stacks previously evaluated. GA, H-GA-1 and H-GA-2 are implemented in a Python 3 environment and computed based on the same parameters (Table 3). Two criteria are implemented to stop the GA: the first stops the algorithm after a fixed number of generations have been created whereas the second stops after a given number of generations without improvement of the best individual in the population. This second criterion assures a certain quality of the best design, even if it is only relative to the previous best design. The comparison criterion is the fitness and the execution time associated with the number of iterations to achieve convergence.

Table 3: Parameters values used in the algorithms

Parameters	Values
Total number of generations	500
Number of individuals per generation	30
Size of chromosomes	8
Probability of mutation	0,2
Mutant gene per individual	1
Probability of crossing	1
Number of individuals chosen for elitism	1
Number of iterations for the local search	5
Number of neighbors per individual	2

Table 4 summarizes the obtained results from a first optimization test where after 500 iterations, the best reached stacking is composed of 6 plies per skin against 8 initially and respects all the design constraints of composite structures thanks to the adopted decoding and repair strategy. Besides both stiffness and buckling constraints are satisfied with margins of +39.2% and +5.9% for GA, H-GA-1 and H-GA-2 respectively. Furthermore, the total mass is decreased by 8.4 % compared to reference. This shows that generally all genetic processes perform well by offering a convincing solution. In fact, the program converges always towards the optimal solution. However, the improvement of the cost of calculation is especially appreciable with H-GA-1 which allows a decrease from 26 to 18 in the number of needed iterations to converge with a gain of 24 minutes on the associated elapsed time. Whereas H-GA-2 fails unfortunately to improve computing time. This can be explained by the fact that dividing the population to undergo a local search greatly expands the number of different individuals processed by the algorithm resulting in wasting resources. Note that H-GA-2 is not retained for further tests.

Table 4: Comparison of results of the first optimization

	Convergence (Iterations/ Time CPU)	Stacking	Stiffness < 2°8 (N/m)	Buckling > 2.03	Mass (Kg)
NASTRAN	-	[45/0/30/0/-30/0/45/90]	1.9°8	2.73	357
GA	26 / 70 min	[30/0/-30/30/-30/90]	1.4°8	2.15	327
H-GA-1	18 / 54 min	[30/0/-30/30/-30/90]	1.4°8	2.15	327
H-GA-2	8 / 165 min	[30/0/-30/30/-30/90]	1.4°8	2.15	327

A second test was performed in order to increase the search space of the algorithms where GA and H-GA-1 are able now to vary the thickness of the sandwich 'core where one individual is encoded and decoded as mentioned earlier (Table 2). The neighborhood structure used in the local search is the same first structure described above but by adding the last gene of the core thickness to the chromosomes. Hence a second optimization is pursued where algorithms run for a total of 100 iterations. Table 5 recapitulates the obtained results; the two methods converge towards a satisfactory solution: they minimize the mass till -10.4% compared to the reference and succeed in reducing the number of plies per skin besides the core thickness while respecting the mechanical and conceptual constraints thanks to the decoding and repair strategy. Furthermore, performances of the H-GA-1 in terms of calculation time are once again proven which highlight the contribution of the hybrid approach against the classical GA.

Table 5: Results of the second optimization

	Convergence (Iterations/ Time CPU)	Stacking	Stiffness < 2°8 (N/m)	Buckling > 2.03	Mass (Kg)
NASTRAN	-	[45/0/30/0/-30/0/45/90/0.22]	1.9°8	2.73	357
GA	16 / 84 min	[90/45/-45/30/-30/0/0.20]	1.6°8	2.04	320
H-GA-1	5 / 30 min	[90/45/-45/30/-30/0/0.20]	1.6°8	2.04	320

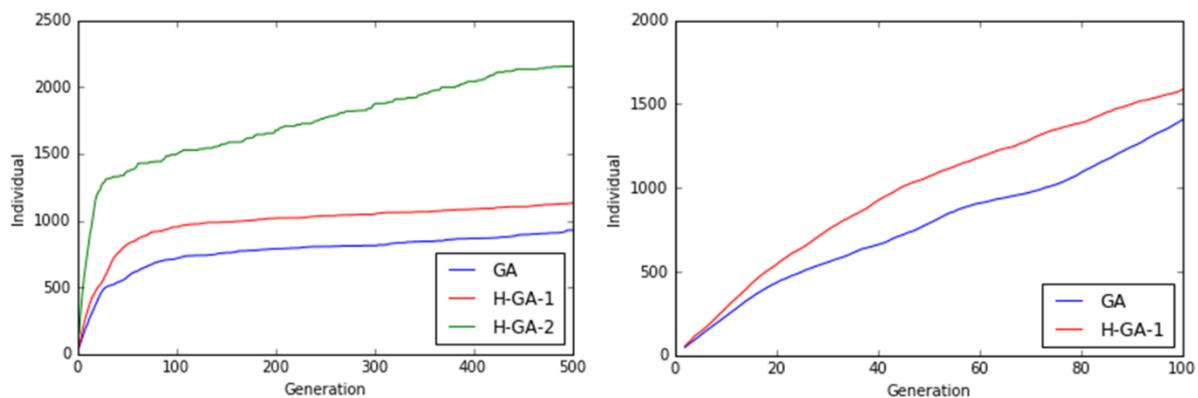


Figure 7 : Evolution of new individuals seen by the algorithms during : (left) the first optimization process, (right) the second optimization

Finally, the graph Figure 7 perfectly illustrates the high usefulness of both based-neighborhood local searches. Indeed, H-GA-1 and H-GA-2 converge by spreading better individuals in the population and offering simultaneously a wide choice of new different individuals to explore thanks to the research intensification around the best solution than that given by the GA. The same result is obtained when the core thickness is integrated besides fiber orientations in the design variable.

## 6. Conclusions

In this paper, we have shown how a hybrid genetic algorithm can outperform the standard genetic algorithm on a sandwich composite structure optimization problem. In the proposed approaches, the population was treated once in a sequential manner resulting in the algorithm H-GA-1 and once by being splitted between genetic operators and descent local search resulting in the algorithm H-GA-2. During a first optimization test, the first approach has greatly improved the quality of the results as well as the speed of resolution and the cost of calculation while the second failed to compute the classical GA. Then, thanks to the capacity of the genetic algorithm to model both discrete and continuous variables, a second optimization test including the sandwich's core thickness among the design variable was performed and the obtained results reveal a better convergence of H-GA-1.

Future work will extend the analysis by integrating fiber materials among the design variables besides testing other local search techniques that might improve the genetic algorithm behavior and therefore the quality of results notably on other test cases.

## References

- [1] T. Reinhart, "Overview of Composite Materials," in Peters S.T. (eds) Handbook of Composites, Boston, MA, Springer, 1998.
- [2] J. I. Pelaez, J. A. Gomez-Ruiz, J. Veintimilla, G. Vaccaro and P. Witt, "Memetic Computing Applied to the Design of Composite Materials and Structures," Mathematical Problems in Engineering, vol. 2017, p. 16.
- [3] A. Rama Mohan Rao and N. Arvind, "Optimal stacking sequence design of laminate composite structures using tabu embedded simulated annealing," Structural Engineering and Mechanics, vol. 25, no. 2, p. 239–268, 2007.
- [4] A. Carpentier, "Optimisation multi-niveaux de panneaux Composites," 2008.
- [5] A. Lasseigne, "Optimization of variable-thickness composite structures. Application to a CROR blade," 2016.
- [6] F.-X. Irisarri, A. Lasseigne, H. Leroy François and R. Le Riche, "Optimisation des empilements dans les structures composites stratifiées avec arrêts de plis," in 18èmes Journées Nationales sur les Composites, Nantes, 2013.
- [7] E.-G. Talbi, Metaheuristics: from design to implementation, New Jersey: John Wiley and Sons Hoboken, 2009.
- [8] C. Vanaret, "Hybridization of evolutionary algorithms and interval-based methods for optimizing difficult problem," 2015.
- [9] N. El Hami, "Contribution to hybrid heuristic optimization methods: Distribution and application on information systems interoperability," 2012.
- [10] C. Blum, M. Aguilera, A. Roli and M. Sampels, "Hybrid Metaheuristics – An Emerging Approach to Optimization," Studies in Computational Intelligence, Springer, vol. 114, 2008.
- [11] J. Hernandez, "Algorithmes Métaheuristiques hybrides pour la sélection de gènes et la classification de données de biopuces," Thèse de doctorat, Université d'Angers, 2008.
- [12] L.M. Schmitt, 2001, "Theory of genetic algorithms", Theoretical Computer Science, Vol. 259, pp. 1-61.
- [13] N. Kogiso, L.T. Watson, Z. Gürdal et R.T. Haftka. « Genetic algorithms with local improvement for composite laminate design ». Structural Optimization, 7(4) :207–208, 1994.
- [14] V.B. Gantovnik, Z. Gürdal et L.T. Watson. « A genetic algorithm with memory for optimal design of laminated sandwich composite panels ». Composite Structures, 58 :513–520, 2002.
- [15] C.-C Lin et Y.-J. Lee. « Stacking sequence optimization of laminated composite structures using genetic algorithm
- [16] Girard, F., 2006, "Optimisation de Stratifiés en Utilisant un Algorithme Génétique", Master's Dissertation, Faculté des Sciences et Génie, Université Laval, Québec, Canada.
- [17] M. Software, MSC Nastran 2012 Quick Reference Guide, M. Software, Ed., 2011.
- [18] J. H. Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence, Ann Arbor, MI: University of Michigan Press, 1975.

- [19] T-Y. Chen, C-J. Chen, Improvements of simple genetic algorithm in structural design. *International Journal for Numerical Methods in Engineering* 1997;40(7):1323-34.
- [20] O. Hasancebi, O. Erbatur, F. Evaluation of crossover techniques in genetic algorithm based optimum structural design. In: Topping BHV, editor. *Advances in engineering computational technology*, Edinburgh, UK: CIVILCOMP Press, 1998:111-123.
- [21] S. Puzzi, A. Carpinteri, 2008, "A double-multiplicative dynamic penalty approach for constrained evolutionary", *Structural and Multidisciplinary Optimization*, Vol. 35, pp. 431-445.
- [22] D. Gay and S.V. Hoa and S.W. Tsai, " *Composite Materials: Design and Applications*," CRC Press, 2002.
- [23] Y. G. Xu, G. Li and Z. Wu, "A novel hybrid genetic algorithm using local optimizer based on heuristic pattern move," *Applied Artificial Intelligence*, vol. 15, no. 7, pp. 601-631, 2001.