

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Hybrid Genetic Algorithms

Leila Gharsalli

Abstract

Hybrid optimization methods have known significant interest in recent years and are being growingly used to solve complex problems in science and engineering. For instance, the famous evolutionary Genetic Algorithm can integrate other techniques within its framework to produce a hybrid global algorithm that takes advantages of that combination and overcomes the disadvantages. Several forms of integration between Genetic Algorithms and other search and optimization techniques exist. This chapter aims to review that and present the design of a hybrid Genetic Algorithm incorporating another local optimization technique while recalling the main local search methods and emphasizing the different approaches for employing their information. A test case from the aerospace field is presented where a hybrid genetic algorithm is proposed for the mechanical sizing of a composite structure located in the upper part of a launcher.

Keywords: genetic algorithm, evolutionary algorithm, hybridization, local search, mechanical sizing, aerospace field

1. Introduction

Solving an optimization problem consists in exploring a search space in order to maximize (or minimize) a given objective function. The complexities (in size or structure) of the search space and the function to be optimized lead to the use of radically different resolution methods. However, the so-called global optimization methods are generally the best suited ones considering their main advantage: that of identifying the promised regions and converging towards a global optimum.

The zero-order optimization Genetic Algorithm (GA), initially laid down by Holland [1], remain the most recognized and practiced form of Evolutionary Algorithms (EA) which are global stochastic optimization techniques that mimic Darwin's principles of natural selection and genetic dynamics. In fact, known advantages of the GA include its ability to combine both exploration and exploitation in an optimal way [1] in addition to its ability to be used in the case of discontinuous objective functions, within disjointed and/or non-convex design spaces, and together with discrete, continuous, or even mixed design variables. Besides, as a population-based method, GA, if initially well-tuned, minimizes the risk to converge to a local optimum thanks to the simultaneous processing of the whole candidate solutions instead of, for instance, gradient-based methods, giving thus the designer a multitude of options.

Nevertheless, like any optimization technique, GA has also its drawbacks starting with the numerous parameters sensitivity analysis to do to maximize efficiency and have a good initialization of the algorithm. In addition to the most notable drawback which is generally the great number of required iterations, and they

slow convergence, especially in the neighborhood of the global optimum. Indeed, although GAs can rapidly locate the region in which the global optimum exists, they take a relatively long time to locate the exact local optimum in the region of convergence. Each generation of the algorithm consists of many objective function evaluations resulting in a lot of computational time in addition to being very expensive.

To overcome this main limitation and lessen function evaluation time, different alternatives have been proposed. But particularly the trend of hybridization has been observed in many works carried out on metaheuristics over the past 20 years, where it consists of incorporating a faster Local Search (LS) (or individual learning) optimization algorithm into a GA in order to improve the performance of the global resulting method, often known as Memetic Algorithm (MA). This latter was first introduced by Moscato [2] and is representing one of the recent growing areas of research in evolutionary computation [3] by offering the important challenge of the tradeoff between global searching and local searching in terms of time and computational effort. In fact, this integration aims to keep advantages of both optimization methods while offsetting their both disadvantages [4]. As LS algorithms are actually capable to find the local optimum with high accuracy and fast convergence but suffer from the problem of foothills. Hence through their integration with the population-based method, the time needed to reach the global optimum can be further reduced since the local knowledge is used to accelerate finding the most promising search region in addition to locating the global optimum starting within its attraction basin.

These hybrid algorithms have also been used under the name of hybrid evolutionary algorithms, Baldwinian evolutionary algorithms, Lamarckian evolutionary algorithms, cultural algorithms, or genetic LS. They also have been successfully applied to hundreds of real-world problems in a wide variety of domains ranging from aircraft [5], aerospace [6], pattern recognition [7], control systems [8], vehicle routing [9], pharmaceutical industry [10] etc.

An example of a hybrid GA applied to the mechanical structures' optimization is presented in this chapter. This test case reflects the relevance of this hybridization in the mathematical resolution of optimization problems related to the aerospace field.

The organization of the chapter is as follows: the second section is dedicated to the description of the different hybridization design including the combination between GAs and LS methods. General mechanisms of GAs are remained in the third section. Hybrid GAs are discussed in the fourth section then the test case with the problematic presentation and the obtained results is given in the fifth section. Finally, conclusions are given in the last section.

2. Hybridization design

As mentioned before, in an optimization problem solving approach, it can be extremely beneficial to combine two solving techniques, knowing that it is possible to hybridize all exact and metaheuristic techniques. Hence, the hybridization can take place in one or more components of a research method. It can also consist in assembling several hybridization methods to form a single hybrid method. Two strategies can be used depending on the optimization technique which drives the hybrid method:

Strategy 1: we can approach the global optimum by a stochastic technique and then refine the result by successively applying a local one. In this case, the result will be ameliorated but unfortunately time consuming. An optimal way to use

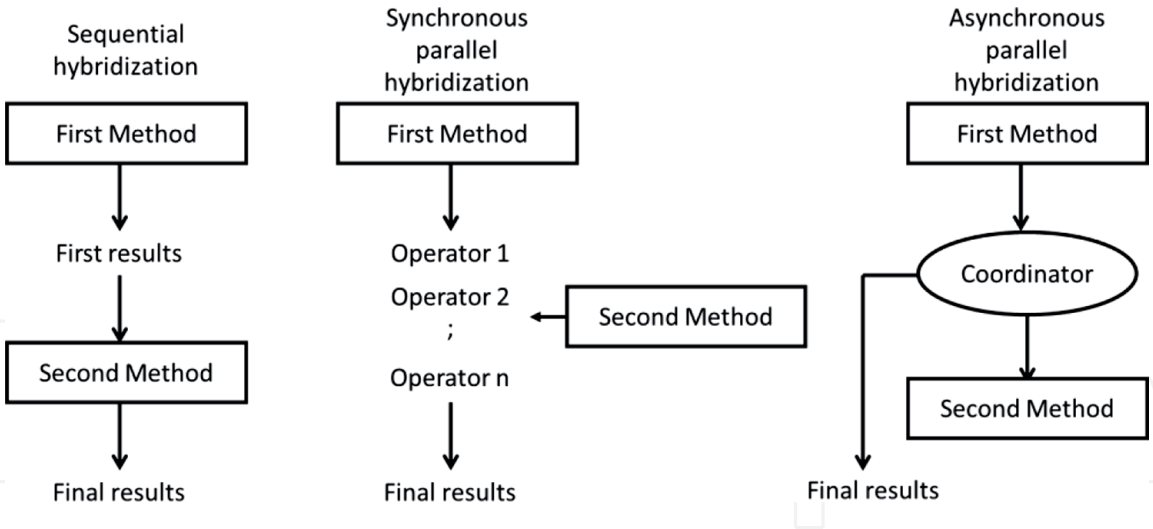


Figure 1.
Hybridization classification.

hybridization in this case may be by applying a global optimization method and find the right time to switch to a local one.

Strategy 2: we can try to find a local minimum and keep it as a winning solution if it is the best of all the local (global) minimums. For instance, in the method of multiple initializations or (multi Start), a local optimization technique is used several times at different stress points; the solution of the optimization problem would be then the best obtained result.

According to the author in [11], a taxonomy for hybrid meta-heuristics can be proposed by classifying them into three categories depending on their architecture (see **Figure 1**):

- Sequential hybridization: it is the most popular type; it performs different research methods sequentially so that the result of the first one serves as an initial solution for the next one.
- Synchronous parallel hybridization: this hybridization is carried out by incorporating a particular research method in an operator. It is more complex to

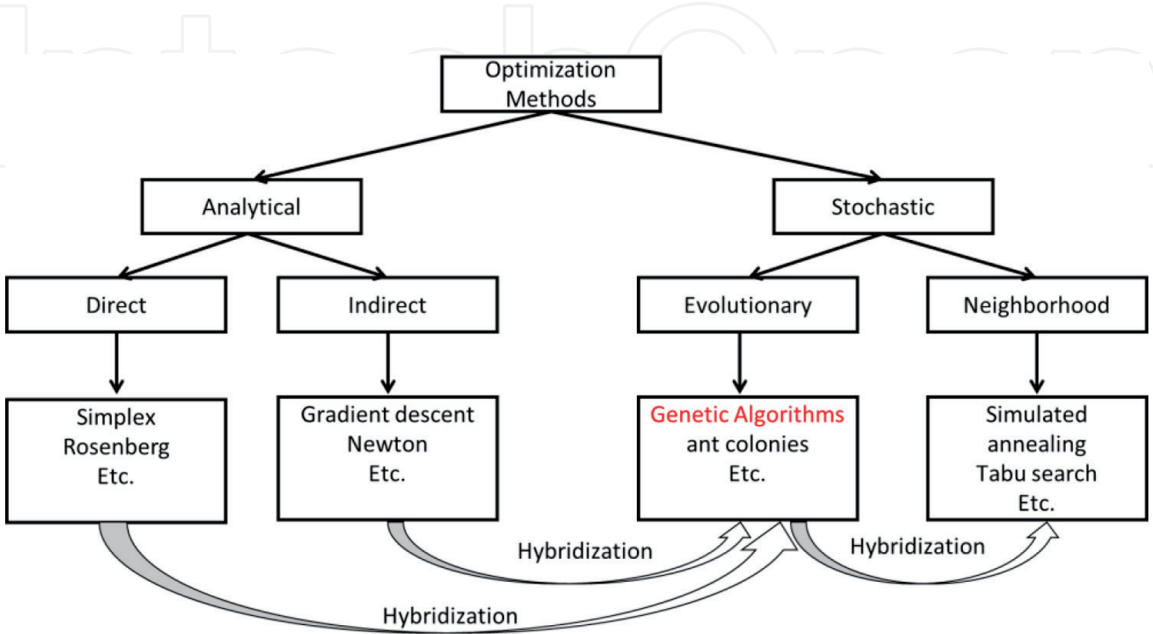


Figure 2.
Possible hybridization schemes.

implement than the previous one. The aim is to combine a LS with a global search for the purpose of improving convergence.

- Asynchronous parallel hybridization: the hybrid methods belonging to this class are characterized by an architecture such that two algorithms A and B are involved simultaneously, and each one adjusts the other. The Algorithms A and B share and exchange information throughout the research process.

A global view of the different possible hybridization schemes is shown in **Figure 2** where we can see hybridization between stochastic (particularly meta-heuristics) and analytical optimization methods. Indeed, the common point between all these hybridizations is that they try to merge the strengths and eliminate the weaknesses of the different concepts. Therefore, the efficiency of the search solution space can still be improved, and new opportunities emerged which may lead to even more powerful and flexible research methods.

3. Genetic algorithm

GAs use a vocabulary inspired from natural genetics. However, the natural processes to which they refer is more complex [12]. They are iterative optimization procedures that repeatedly apply GA operators (such as selection, crossover, and mutation) to a group of solutions until some criterion of convergence has been satisfied. Note that the essential element of the genetic vocabulary is the “individual” that belongs to a set of individuals called “population”. The individual is made up of a “chromosome” which is itself made up of “genes” which contain the inheritance characteristics of the individual. The principles of “selection”, “crossover” and “mutation” are the main genetic operators that are inspired by the same natural processes. First, the selection operator includes generally two variants; the parental selection devoted for reproduction and the replacement selection devoted to keeping the population size constant by individuals who will survive in the next generation. A selection favors the best individuals but has also to give a chance to the less good to avoid the problems of premature convergence. Note that the “fitness” or “evaluation” function characterizing how well adapted the solution is to its environment, is closely related to the selection process. Then, the crossover operator involves combining the information from two parents to create one or two new individuals (or “offspring”) making it possible to diversify the population and explore more the research space. At last, the mutation operator causes a small perturbation on an individual’s chromosome thus allowing to guarantee gene diversity so that the algorithm does not get stuck in local minima. This process continues, throughout the generations, until the stop criteria is reached. The latter can be in various forms such as for example several global iterations defined in advance or the difference between the average population’s fitness and the best one.

It is also important to underline that during the implementation of the algorithm it is necessary to take into account the encoding of the different parameters. In fact, one of the characteristics of GAs is their need to a genetic representation of the desired problem using an encoding. Therefore, the algorithm can evaluate the fitness of every individual and computes the problem’s objectives and constraints based only on the encoding without any further knowledge of real parameters.

Figure 3 shows the optimization flowchart offered by GA whereas Reference [13] provides further discussion on the theoretical properties of GAs and on standard genetic operators.

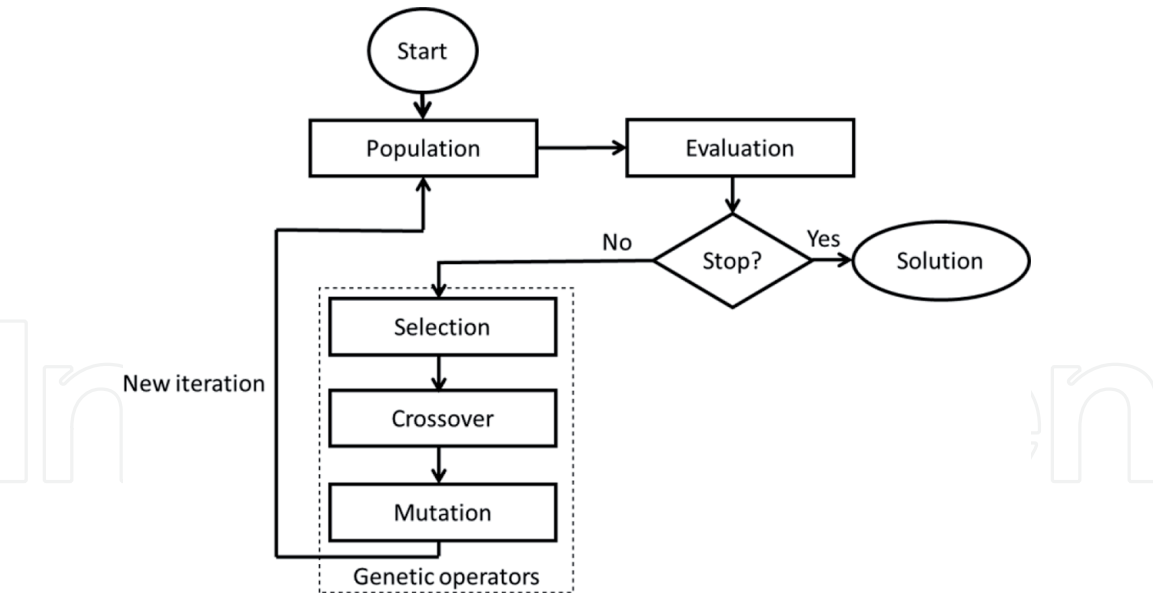


Figure 3.
Flowchart of the optimization procedure based on a genetic algorithm.

4. Hybrid genetic algorithm

The most natural hybridization in the literature is the use of a single solution meta-heuristic algorithm embedded in a population-based meta-heuristic algorithm like the unmissable GA in order to improve the qualities of the solutions. In fact, whereas population-based meta-heuristics tend to diversify the search space by exploring its different parts, LS meta-heuristics tend to intensify the search by exploiting only one part of the space.

LS methods are based on a neighborhood relation and on a procedure exploiting this neighborhood. They consist in moving from one solution to another closer in the space of the candidate solutions, until a solution considered as optimal is found, or the allowed time is exceeded. We can therefore summarize the main idea of a LS as follows:

- Start from an initial solution,
- Improve that solution,
- Repeat the process several times.

Hence, through the hybridization, individuals attempt to identify promising areas of the solution space which are then explored in more detail by LS methods. The best known among the latter are the simulated annealing, the tabu search or simply descent methods (iterated local searches).

Concerning the hybridization adopting the global GA, as its decision-making mechanics are based on random principles, this detailed research can be difficult to carry out. Heuristics therefore serve to fill this weakness. In a general way, at all generations, an individual (or some individuals) are randomly selected among the best in the population. Then, a set of solutions defining its (their) neighborhood is constructed. Afterwards, each of these neighbors is evaluated in order to check whether there is an improvement in the evaluation function. GA then manipulates solutions that have already been improved locally. This has the effect of allowing even more advantageous solutions to be generated more quickly. However, as there are several ways to hybridize GAs while maintaining a fairly modular program structure, a sequential approach could be advantageous in such a way that it is

enough to let the GA run to a substantial level of convergence, then let the local optimization procedure follow up the process, for example by taking the 5% or the 10% best individuals of the last generation. However, two main approaches are also often followed: firstly, the Lamarckian approach [14] where direct learning transmits the best characteristics of each individual from generation to generation. Hence, both change in genotypic information and fitness are transmitted to the individual as genotypic information at the end of the LS, altering thereby the chromosomes. Secondly, the Baldwinian approach [14] where only the improved fitness function value is changed after the LS and not the genotypic information. The first approach is known to be faster than the second one but may cause premature convergence [13].

Note that usually the individuals that undergo LS are chosen uniformly at random, but it would be more efficient to use a tuning technique which consists of a primary conducted experiment aiming to find the optimal part of the population that should perform LS. This part is commonly referred to as the LS probability mentioned above that is used to run the actual experiment and remains fixed during the algorithm execution. In the literature, several techniques aim to reduce unnecessary local optimization and therefore additional calculation time, such as distribution-based techniques [15], fitness-based techniques [15] and LS potential [16] that have been proposed to select the optimal individuals among the given population that should do a LS. In the following, some main LS methods are remained.

4.1 Simulated annealing (SA)

Simulated annealing is an optimization technique inspired by the simulation methods of Metropolis (1950s) in statistical mechanics. It is distinguished by the introduction of a temperature parameter which is adjusted during research [17]. Moreover, in optimization, this method considers a neighborhood exploitation procedure which makes it possible to go towards a neighboring solution of less good quality with a non-zero probability, thus allowing to escape the local optima.

At the beginning, a “temperature” parameter T is determined and decreases throughout the algorithm to tend towards 0. The probability of accepting deteriorated solutions depends on the value of this parameter (the more the temperature T decreases, the more this probability decreases). The interest of simulated annealing lies in the fact that there is a proof of its asymptotic convergence. Thus, when certain conditions are verified (decrease diagram of T), we are guaranteed to obtain the optimal solution. However, the parameterization recommended by theory is not very realistic and it takes a long time to get to parameterize these methods. Note that this method may also require a stop criterion, if the “optimal” setting has not been found. Hybridization between GA and SA algorithms has been used in several domains [18–20].

4.2 Tabu search (TS)

Tabu search was introduced as a new strategy to escape local optima using a notion of memory [21]. Exploitation of the neighborhood makes it possible to move from the current solution to its best neighbor (this one not necessarily having a better quality than the current solution). To avoid cycling between a local optimum and its best neighbor, the method prohibits moving to a recently visited solution. To do this, a tabu list containing the attributes of the last visited solutions is kept up to date. Each new solution considered removes the oldest visited solution from this list. Thus, the search for the following current solution is done in the neighborhood of the current solution without considering the solutions belonging to the tabu list.

However, the size of this list is a method parameter that is difficult to adjust. In addition, this strategy requires the definition of a stopping criterion. We find the hybridization between GA and TS in many applications such as image processing [22], computational grids [23] and flow shop scheduling problem [24].

4.3 Iterated local search (ILS)

The descent methods are quick and simple to implement but generally do not lead to the best optima because they stop as soon as a local optimum is found [25]. In order not to get stuck on this local optimum, there are various strategies which allow the search to continue after having found an optimum. One strategy to overcome the abrupt stopping of the search for a local optimum is to iterate the descent method. The following steps are carried out from the found optimum:

- Apply a perturbation on the current solution,
- Apply a descent method on that solution.
- Chose via an acceptance criterion if the new optimum becomes the current solution and go back to the first step until the stop criterion is reached. Common stop criteria are the execution time, the number of iterations and the number of total evaluations.

The disturbance can consist in restarting a solution taken randomly in the search space or in choosing a solution in a neighborhood far from the optimum or even in choosing a neighbor of the same quality as the optimum.

Heuristic Iterative Local Search (ILS) [26] is based on a simple idea: instead of repeatedly applying a LS procedure from randomly generated solutions, ILS generates the starting solution for the next iteration by applying a perturbation on the local optimum found at the current iteration. This is done in the hope that the disturbance mechanism provides a solution located in the attraction basin of a better local optimum. Therefore, the disruption mechanism is a key part of ILS. In addition, an acceptance criterion defines the conditions that the new local optimum must meet in order to replace the current local optimum (**Figure 4**). Thus, the acceptance criterion, combined with the disruption mechanism, helps to control the trade-off between intensification and diversification. However, many acceptance

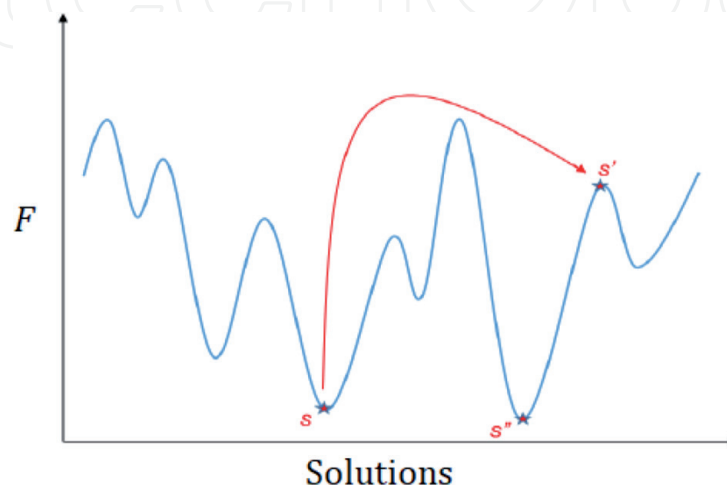


Figure 4.
 From a local minimum s , a disturbance of the latter generates a minimum s' from which a LS is launched to arrive at a third local minimum s'' potentially better than s .

criteria which reconcile the two objectives can be applied [12]. Hybridization between GA and ILS is one of the simplest and common hybridizations that can be found in several works [27–29].

4.4 Variable neighborhood search (VNS)

The Variable Neighborhood Search (VNS) is a meta-heuristic algorithm [30], based on the principle of systematic neighborhood change during LS. Indeed, at the initialization step, a set of neighborhood structures must be defined. These neighborhoods can be chosen arbitrarily, but often a sequence ordered in size growing neighborhoods is used. The VNS procedure consists of three stages: (1) the disturbance (shaking), (2) the LS and (3) the displacement. Recently, a variety of problems have involved hybridization between GA and VNS [31–33].

5. Test case

We consider here the following optimization problem: search for the minimum mass of an inter-stage skirt made of composite sandwich with Carbon Fiber Reinforced Polymer (CFRP) skins and aluminum Nida core (composites are fabricated by attaching two thin but stiff skins to a lightweight but thick core) that are located between the first stage and the second stage situated on the upper part of a linear launcher (**Figure 5**).

The structure is subjected to pressure oscillations from upstream solid propellant engines to ensure “payload comfort”. The skirt therefore has a filtering role in order to limit the vibratory levels at the top of the launcher. To best dampen these oscillations, the structure must be mechanical sized according to a compromise between rigidity and flexibility linked to the need to hold in buckling, in addition to composite manufacturing constraints (mirror symmetry, balancing, grouping) [34]. Note that we need to optimize only one skin plus the core hence the final result is obtained by symmetry.

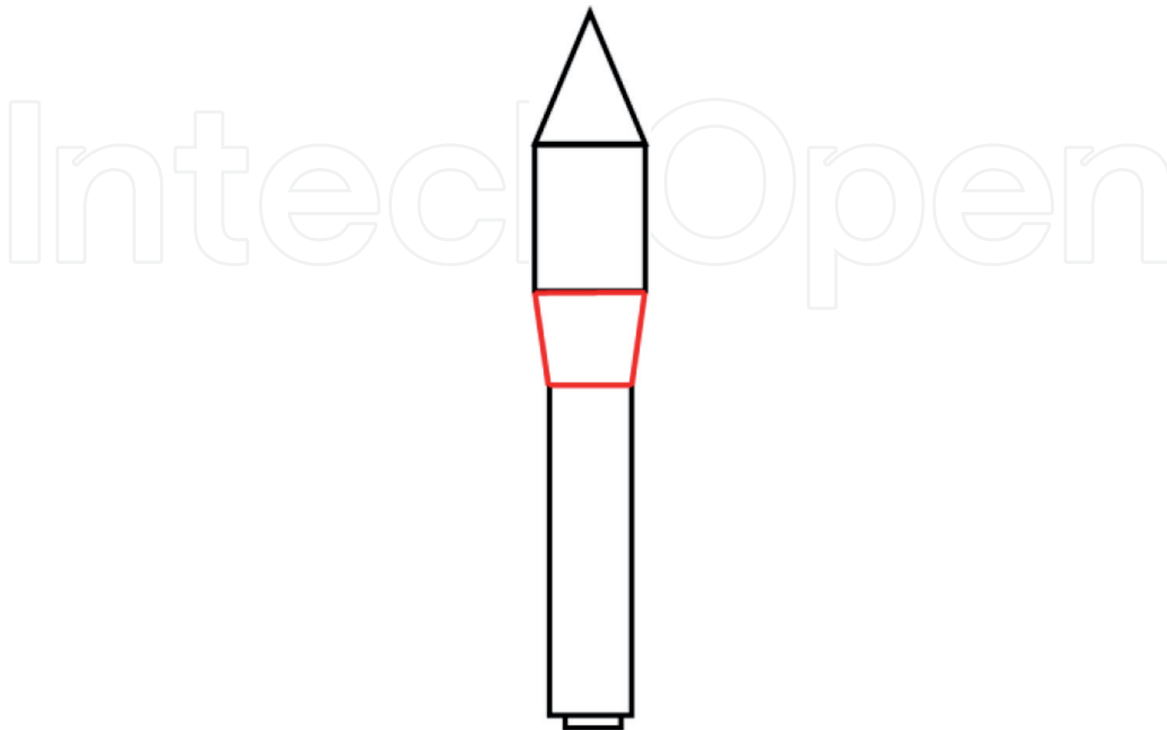


Figure 5.
Simplified model of the sandwich composite inter-stage skirt (in red) located on the upper part of the linear launcher.

Mathematically speaking, we can write the optimization problem as follows:
Objective: minimize the total mass (M).
 Subject to:

$$\text{Constraints: } \begin{cases} \text{structure's buckling } (B)(> 2,03) \\ \text{structure's stiffness } (S)(< 2e8 \text{ N / m}) \\ \text{fabrication constraints } (FC) \end{cases}$$

$$\text{Design variables: } \begin{cases} \text{total number of plies forming the skin} \\ \text{plies thicknesses} \end{cases}$$

5.1 Proposed hybrid genetic algorithm

To solve the above optimization problem, we have chosen the global GA. In fact, during the mechanical sizing of a composite structure, plies thicknesses are often predetermined, and plies orientations are usually restricted to a small set of angles due to manufacturing constraints and/or limited availability of experimental data. Most of the time, this gives rise mathematically to a discrete optimization problem. Hence GA is a suitable optimization method that we also decided to hybridize with a LS method to overcome the drawbacks of GA cited throughout this chapter.

First of all, an appropriate representation of the problem is recommended through the encoding of all the parameters. In fact, the initial population of the GA is composed of a set of skin plies (sandwich case) characterized by fiber orientations. Thus, an individual in the population is equivalent to a laminate (set of plies) and can simply be represented by an orientation chromosome (see **Figure 6**). Plies and core thicknesses of the structure are predefined based on composite design constraints [35] and are not considered herein as optimization variables. Therefore, an individual (a laminate) is represented by one chromosome regrouping one skin plies characterized by orientation angles (0° , 90° , $\pm 45^\circ$, $\pm 30^\circ$), encoded according to 5 possible values $\{0, 1, 2, 3, 4\}$ called the chromosome domain where particularly add where 0° is encoded by 0, 90° is encoded by 1, $\pm 45^\circ$ is encoded by 2, “no ply” is encoded by 3 and finally $\pm 30^\circ$ is encoded by 4. “no ply” is encoded by 3.

To handle constraints, concerning the composite manufacturing ones, a repair strategy is adopted during the decoding phase (**Figure 7**). More explanation about these constraints handling method could be seen in [36]. Whereas mechanical constraints (buckling and stiffness) are managed through the genetic selection operator.

The proposed Hybrid Genetic Algorithm (HGA) herein is based on a strong and sequential cooperation between GA and a descent LS technique for the entire course

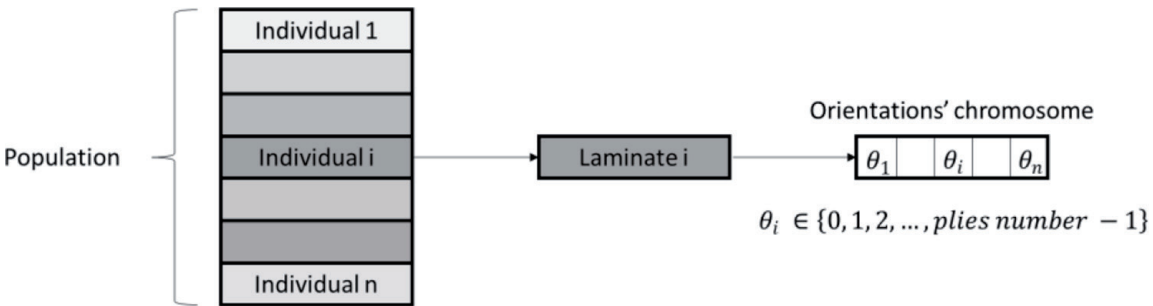


Figure 6.
 The genetic representation of the design variables.

of the algorithm where the LS optimization is integrated at the beginning of the GA previous to the genetic operators to intensify research only around selected promising solutions. Hence, the adopted scheme is as follows: for every new iteration, the best individual reached by the GA is selected as an initialization for the LS method. Then, a set of two neighbors is built following a well-defined structure. The latter aims to verify if there is any neighboring orientation that allows an improvement of the individual's fitness by randomly selecting a gene related to the orientation and flipping it to either the previous or the next value in the concerned chromosome's domain while of course respecting all manufacturing constraints via the repair cited repair strategy. For example, if the selected gene's code is 2, the value of this gene will be replaced by 1 in the first neighbor and by 3 in the second one. However, if the randomly selected gene is the domain's upper bound (4) or the lower one, then the two neighbors are built by flipping firstly the gene to the previous value in the domain (3) then to the lower bound (0) or conversely.

Afterwards, each neighbor is evaluated and if one's fitness turns out to be better, then it is introduced in the population to replace the first individual and so on **Figure 7** shows an example of the decoding and repair strategy where the steps decoding then repairing are explained, the randomly selected gene's code is 2, corresponding to 90°. In order to respect the balancing composite rule, that gene should

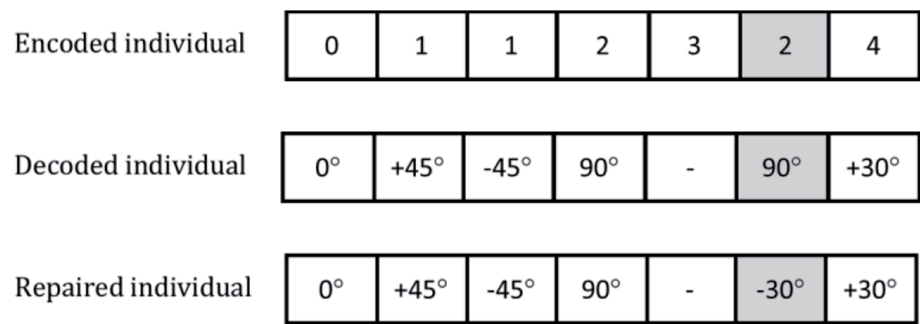


Figure 7.
Example of the decoding and repair strategy.

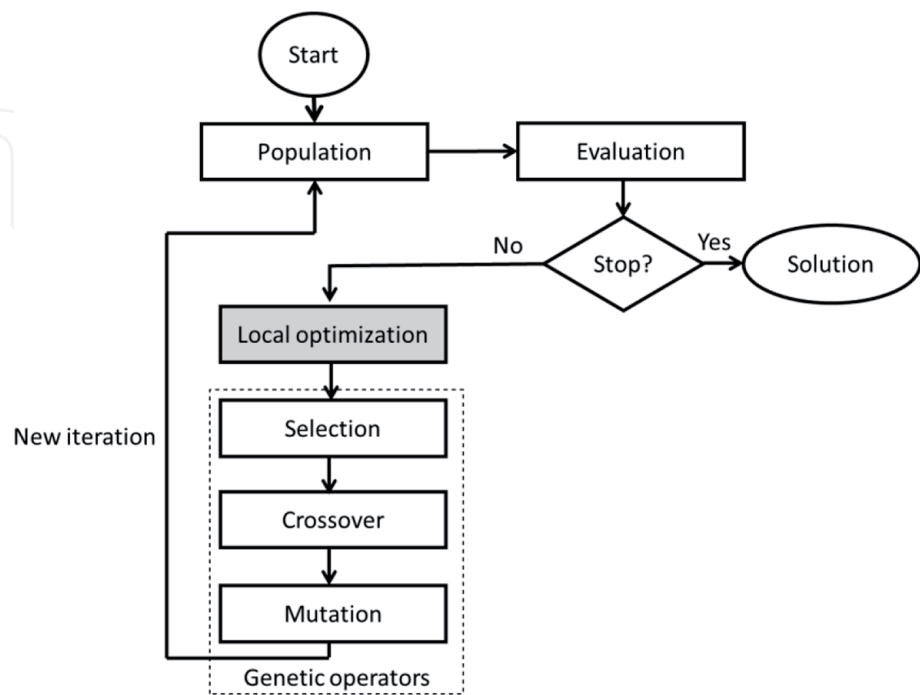


Figure 8.
Flowchart of the optimization procedure based on the hybrid genetic algorithm (HGA).

Parameters	Values
Total number of iterations	100
Number of individuals per generation	30
Chromosomes' size	8
Mutation probability	0,2
Mutant gene per individual	1
Crossover probability	1
Number of individuals chosen during the selection	1
Number of iterations for the LS	5
Number of neighbors per individual	2

Table 1.
Initialization of the algorithms' parameters.

be replaced by -30° (repair) so the entire laminate respects the composite design constraints. The **Figure 8** shows the adopted HGA flowchart.

5.2 Application and results

A comparative study between the classical GA and the proposed HGA is presented in the following. GA and HGA are implemented in a Python 3 environment and computed based on the same parameters (see **Table 1**). Note that at each iteration, the algorithms run through a database containing individuals already assessed with their respective fitness calculated through the finite element analysis program called NASTRAN [37] and look for the latter from the stacks previously evaluated. Besides, two criteria are implemented to stop the running of 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. Concerning the HGA, the stop criteria is a prior fixed number of iterations.

Table 2 summarizes the optimization results. After 100 iterations, the best reached laminate 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 and HGA, 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,

	Laminate (°)	S < 2°8 (N/m)	B > 2.03	M (Kg)	Convergence (Iterations/Time CPU)
NASTRAN	[45/0/30/0/-30/0/45/90]	1.9°8	2.73	357	—
GA	[30/0/-30/30/-30/90]	1.4°8	2.15	327	26/70 min
HGA	[30/0/-30/30/-30/90]	1.4°8	2.15	327	18/54 min

Table 2.
Results of the comparative study between GA and HGA.

the program converges always towards the optimal solution. However, HGA provides 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. Thereby, HGA can outperform the standard GA on this sandwich composite structure optimization problem which highlight the hybridization mechanism and its capacity in the improvement of convergence.

6. Conclusion

In this chapter we have underlined the importance of the process of hybridization in ameliorating the global genetic algorithm behavior and especially its convergence. In fact, although genetic algorithms are effective complete search algorithms able to combine both exploration and exploitation with crossover and mutation operators, they suffer from being computationally expensive and can hence be improved using local search methods, so they can be made competitive with other algorithms when the search space is too large to explore. More generally, the combination of metaheuristics based on population as well as on local search, may offer more effective and dynamic methods for the solution of several problems.

The following important questions were asked: (1) when to apply local search optimization (2) to which individuals in the genetic algorithm should local searches be applied, and (3) what are the possible hybridizations that could improve the optimization results. Some answers to these questions have been provided particularly the reminder of the hybridization strategies and its different possible architectures in addition to the main local search methods known by their relevant coupling with the classical genetic algorithm. The latter being able to undergo different forms of hybridization thanks to its flexibility which is nevertheless one of its strongest points.

The application possibilities of these approaches are unlimited but in this chapter the test case that was presented is about an optimization algorithm combining traditional genetic operators with an iterated local search method specially designed to pursue the problem of mechanical sizing of a composite structure. The genetic algorithm was chosen as the global optimization tool because of its ability to deal with non-convex and discrete optimization problems, of which the design of laminated composites is an example. Besides, its flexible structure makes it possible to integrate the management of different mechanical and manufacturing constraints through genetic operators and the implementation of specific optimization processes such as the decoding and repair strategy. The developed hybrid algorithm was validated by comparing its results to those obtained via the classical genetic algorithm as well as a reference finite element analysis software. That test case showed hence that the developed hybrid genetic approach is able to obtain efficient results in a real-world problem related to the aerospace sector.

Acknowledgements

The author wishes to thank the French National Center for Space Studies that partially supported this work.

Conflict of interest

The authors declare no conflict of interest.

IntechOpen

IntechOpen

Author details

Leila Gharsalli
Polytechnic Institute of Advanced Sciences (IPSA), Ivry-sur-Seine, France

*Address all correspondence to: leila.gharsalli@ipsa.fr

IntechOpen

© 2022 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Holland JH. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. MI: University of Michigan Press; 1975
- [2] Moscato PA. Memetic algorithms: A short introduction. In: Corne D, Dorigo M, Glover F, editors. *New Ideas in Optimization*. London: McGraw-Hill; pp. 219-234; 1999
- [3] Chen X, Ong Y, Lim M, Tan KC. A multi-facet survey on memetic computation. *IEEE Transactions on Evolutionary Computation*. 2011;15(5):591-607. DOI: 10.1109/TEVC.2011.2132725
- [4] Wan W, Birch JB. An improved hybrid genetic algorithm with a new local search procedure. *Journal of Applied Mathematics*. Virginia Tech Publisher. Vol. 2013, p.10. Article ID 103591. DOI: 10.1155/2013/103591
- [5] Bencheikh G, Boukachour J, El Hilali AA. A memetic algorithm to solve the dynamic multiple runway aircraft landing problem. *Journal of King Saud University-Computer and Information Sciences*. 2016;28(11):98-109
- [6] Gharsalli L, Guérin Y. Mechanical sizing of a composite launcher structure by hybridizing a genetic algorithm with a local search method. *Composites Part C: Open Access*. 2021;5:100125. DOI: 10.1016/j.jcomc.2021.100125
- [7] Montazeri M, Montazeri M, Reza Naji H, Faraahi A. A novel memetic feature selection algorithm. In: *The 5th Conference on Information and Knowledge Technology*. Shiraz, Iran: IEEE publisher; 2013. DOI: 10.1109/IKT.2013.6620082
- [8] Shyr WJ, Wang BW, Yeh YY, Su TJ. Design of optimal PID controllers using memetic algorithm. In: *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)*. Anchorage, AK, USA. Vol. 3. 2002. pp. 2130-2131. DOI: 10.1109/ACC.2002.1023951
- [9] Labadi N, Prins C, Reghioui M. A memetic algorithm for the vehicle routing problem with time windows. *RAIRO-Operations Research-Recherche Opérationnelle*. 2008;42(3):415-431. DOI: 10.1051/ro:2008021
- [10] Tse S-M, Liang Y, Leung K-S, Lee K-H, Mok T-S. A memetic algorithm for multiple-drug cancer chemotherapy schedule optimization. *IEEE Transactions on Systems, Man, and Cybernetics. Part B, Cybernetics*. 2007;37(1):84-91. DOI: 10.1109/tsmcb.2006.883265
- [11] Talbi EG. A taxonomy of hybrid metaheuristics. *Journal of Heuristics*. 2002;8:541-564. DOI: 10.1023/A:1016540724870
- [12] Talbi EG. *Metaheuristics: From Design to Implementation*. Vol. 74. Hoboken, New Jersey: John Wiley & Sons; 2009. DOI: 10.1002/9780470496916
- [13] Whitley D. A genetic algorithm tutorial. *Statistics and Computing*. 1994;4:65-85. DOI: 10.1007/BF00175354
- [14] Julstrom BA. Comparing Darwinian, Baldwinian, and Lamarckian search in a genetic algorithm for the 4-cycle problem. In: *Late Breaking Papers at the 1999 Genetic and Evolutionary Computation Conference*. Orlando, Florida, USA: Morgan Kaufmann Publishers Inc; 1999. pp. 134-138
- [15] Hart WE. *Adaptive Global Optimization with Local Search [Thesis]*. USA: University of California at San Diego; 1994

- [16] Shannon Land MW, K. Belew R. Evolutionary Algorithms with Local Search for Combinatorial Optimization (thesis). CA, USA: University of California, San Diego; 1998
- [17] Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing. In: Fischler MA, Firschein O, editors. Readings in Computer Vision. San Francisco, CA, United States: Morgan Kaufmann publisher 1987. pp. 606-615. DOI: 10.1016/B978-0-08-051581-6.50059-3
- [18] Chen P-H, Shahandashti SM. Hybrid of genetic algorithm and simulated annealing for multiple project scheduling with multiple resource constraints. Automation in Construction. 2009;18(14):434-443. DOI: 10.1016/j.autcon.2008.10.007
- [19] Li Y, Guo H, Wang L, Fu J. A hybrid genetic-simulated annealing algorithm for the location-inventory-routing problem considering returns under E-supply chain environment. The Scientific World Journal. Vol. 2013. p.10. DOI: 10.1155/2013/125893
- [20] Chen D, Lee C-Y, Park C-H. Hybrid genetic algorithm and simulated annealing (HGASA) in global function optimization. In: 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'05). Hong Kong, China: IEEE publisher; 2005. pp. 5-133. DOI: 10.1109/ICTAI.2005.72
- [21] Glover F, Laguna M. Tabu search. In: Handbook of Combinatorial Optimization. Boston, MA: Springer; 1998. pp. 2093-2229
- [22] Haddad M, Jarray F, Tlig G, Hasni H. Hybridization of genetic algorithms and tabu search for reconstructing convex binary images from discrete orthogonal projections. International Journal of Metaheuristics. 2014;3(4):291-319
- [23] Khafa F, Gonzalez JA, Dahal KP, Abraham A. A GA (TS) hybrid algorithm for scheduling in computational grids. In: Corchado E, Wu X, Oja E, Herrero Á, Baroque B, editors. Hybrid Artificial Intelligence Systems. HAIS 2009. Lecture Notes in Computer Science. Vol. 5572. Berlin, Heidelberg: Springer; 2009. DOI: 10.1007/978-3-642-02319-4_34
- [24] Umam MS, Mustafid M, Suryono S. A hybrid genetic algorithm and tabu search for minimizing make span in flow shop scheduling problem. Journal of King Saud University-Computer and Information Sciences. Vol. 53. 2021. DOI: 10.1016/j.jksuci.2021.08.025
- [25] Papadimitriou C, Steiglitz K. Combinatorial optimization: Algorithms and complexity. Mineola, New York: Dover Publications, Inc.; 1998
- [26] Stützle T. Local search algorithms for combinatorial problems - analysis, improvements, and new applications. Darmstadt University of Technology, Germany, 1999. Vol. 220. pp. I-XI, 1-203. ISBN 978-3-89601-220-3
- [27] Derbel H, Jarboui B, Hanafi S, Chabchoub H. Genetic algorithm with iterated local search for solving a location-routing problem. Expert Systems With Applications. 2012;39(13): 2865-2871. DOI: 10.1016/j.eswa.2011.08.146
- [28] Gharsalli L, Guérin Y. A hybrid genetic algorithm with local search approach for composite structures optimization. In: 8th European Conference for Aeronautics and Space Sciences. Madrid. AIP publisher. 2019. p. 9. DOI: 10.13009/EUCASS2019-550
- [29] Shuli H, Huan L, Xiaoli W, Ruizhi L, Junping Z, Jianan W. A hybrid framework combining genetic algorithm with iterated local search for the dominating tree problem. Mathematics. 2019;7(14):359. DOI: 10.3390/math7040359

- [30] Mladenovic N, Hansen P. Variable neighborhood search. *Computers and Operations Research*. 1997;**24**(111): 1097-1100. DOI: 10.1016/S0305-0548(97)00031-2
- [31] Duda J. A hybrid genetic algorithm and variable neighborhood search for multi-family capacitated lot-sizing problem. *Electronic Notes in Discrete Mathematics*. 2017;**58**:103-110. DOI: 10.1016/j.endm.2017.03.014
- [32] Fuksz L, Pop PC. A Hybrid genetic algorithm with variable neighborhood search approach to the number partitioning problem. In: Pan JS, Polycarpou MM, Woźniak M, de Carvalho ACPLF, Quintián H, Corchado E. (eds). *Hybrid Artificial Intelligent Systems HAIS. Lecture Notes in Computer Science*, vol. 8073. Berlin, Heidelberg: Springer; 2013. DOI: 10.1007/978-3-642-40846-5_65
- [33] Luo Y, Pan Y, Li C, Tang H. A hybrid algorithm combining genetic algorithm and variable neighborhood search for process sequencing optimization of large-size problem. *International Journal of Computer Integrated Manufacturing*. 2020;**33**(110-11): 962-981. DOI: 10.1080/0951192X.2020.1780318
- [34] Mårtensson P, Zenkert D, Åkermo M. Effects of manufacturing constraints on the cost and weight efficiency of integral and differential automotive composite structures. *Composite Structures*. 2015;**134**: 572-578. DOI: 10.1016/j.compstruct.2015.08.115
- [35] Gay D. *Composite Materials: Design and Applications*. Third ed. Boca Raton, Floride, USA: CRC Press; 2014
- [36] Puzzi S, Carpinteri A. A double-multiplicative dynamic penalty approach for constrained evolutionary optimization. *Structural and Multidisciplinary Optimization*. 2008;**35**:431-445. DOI: 10.1007/s00158-007-0143-1
- [37] MSC Nastran 2012 Quick Reference Guide. MSC Software corporation. USA. 2012