J Glob Optim (2010) 48:399–421
DOI 10.1007/s10898-009-9503-2

BIANCA: a genetic algorithm to solve hard combinatorial optimisation problems in engineering

Angela Vincenti · Mohammad Reza Ahmadian · Paolo Vannucci

Received: 3 September 2009 / Accepted: 2 December 2009 / Published online: 20 December 2009

Abstract The genetic algorithm BIANCA, developed for design and optimisation of composite laminates, is a multi-population genetic algorithm, capable to deal with unconstrained and constrained hard combinatorial optimisation problems in engineering. The effectiveness and robustness of BIANCA rely on the great generality and richness in the representation of the information, i.e. the structure of populations and individuals in BIANCA, and on the way the information is extensively exploited during genetic operations. Moreover, we developed proper and original strategies to treat constrained optimisation problems through the generalisation of penalisation methods. BIANCA can also treat constrained multi-objective problems based on the construction of the Pareto frontier. Therefore, BIANCA allows us to approach very general design problems for composite laminates, but also to make a step forward to the treatment of more general problems of optimisation of materials and structures. In this paper, we describe specifically the case of optimal design of composite laminates, concerning both the theoretical formulation and the numeric resolution.

Keywords Genetic algorithms · Genetic encoding · Constraints · Multi-objective optimisation · Structural optimisation · Composite laminates

A. Vincenti (✉) · P. Vannucci
Institut Jean Le Rond d'Alembert, UPMC Univ Paris 06, UMR 7190, Case 161, 4 place Jussieu, 75252 Cedex Paris, France
e-mail: angela.vincenti@upmc.fr

A. Vincenti · P. Vannucci
Institut Jean Le Rond d'Alembert, CNRS UMR 7190, Case 161, 4 place Jussieu, 75252 Cedex Paris, France

M. R. Ahmadian · P. Vannucci
Université de Versailles et Saint Quentin, 35 avenue des Etats-Unis, 78035 Versailles, France

1 Introduction

In the domain of engineering design, a system has to be conceived in order to respond to several criteria which constitute a list of design specifications. Traditionally, in a first attempt to solve a design problem the goal is to simply respond to this specification list, even if the solution appears to be perfectible. The history of engineering shows old design solutions undergoing further improvements in order to perfect the system and eventually optimise it. However, nowadays design specifications are stricter and stricter, and the need appears to reach the perfectible limits of an engineering system in the first go of the design process: the urge is to spare time, material, mass, money. It is therefore more and more common to directly formulate a design problem in the form of an optimisation problem.

Often optimisation problems issued of engineering design are complex in terms of mathematical formulation, as they can be highly non linear and non convex [1–3]. Moreover, the multiplicity of criteria to respect leads to multi-objective optimisation problems, and often with imposed constraints. Finally, the manifold of design variables to be taken into account belongs to different sets: continuous, discrete, and grouped.

The complexity of optimal design in engineering applications is therefore twofold: on one hand, the need is to focus on the formulation for the optimal design problem, in order to make it more and more effective and appropriate; but also a considerable effort has to be made in the development of numerical techniques apt to solve difficult optimisation problems in the most general way, i.e. capable to take into account all criteria and constraints at the same time, and capable to deal with all types of design variables.

In this sense, the development of metaheuristics and of evolutionary strategies, such as genetic algorithms, played a very important role in broadening the scope of optimisation in engineering design [3–5]. Besides the fundamental advantage of being effective in finding global optima when applied to non convex objective functions, these techniques show many interesting qualities. First of all, they have a quite free and simple programming scheme, which allows tailoring the right strategy in order to suit the needs issued from different design problems.

Nevertheless, the simplicity of the programming scheme does not imply any simplification in the description of the design space: on the contrary, the effectiveness of metaheuristics relies also on the rich and exhaustive representation of parameters and variables for the design problem.

All these considerations were fundamental when we had to decide which strategy we could follow to solve design problems for composite laminated plates, i.e. plates composed of a number of fiber-reinforced composite layers stacked according to a sequence of orientation angles [6].

Composite design problems are hard combinatorial optimisation problems, where a high number of design parameters must be defined in order to obtain given properties and/or to optimise some laminate behaviors. Design variables can be of various types, but in real-world design problems they are essentially discrete or grouped. Moreover, all design parameters have to be taken into account simultaneously in order to assure the greatest generality of the design process.

It appears clearly here that a key issue is the representation of the information, which has to be detailed but essential, rich and exhaustive. In this sense, the way genetic algorithms deal with information suits very well our problems, since we could develop a complete and detailed representation of a composite laminate in the form of genes and chromosomes.

Nevertheless, it is important to point out that our choice of a genetic algorithm as a global numerical technique to solve composite laminate design problems is also based on the very nature of our formulation, which is in the form of a highly non linear and non convex optimisation problem.

Vincenti and Vannucci created BIANCA, a genetic algorithm for design and optimisation of composite laminates [7]. Based on the structure of a standard genetic algorithm, BIANCA already showed some relevant points of originality in the representation of the genotype for composite laminates and a new strategy in the treatment of inequality constraints [7,8]. We performed a large campaign of numerical tests using BIANCA, which proved to be a very effective and robust tool to deal with optimal design of composite laminates.

Nevertheless, further improvements and enrichments in our formulation of design problems for composite laminates led us also to renew and enrich the genetic algorithm BIANCA, as we describe in this paper.

Nowadays, BIANCA is a multi-population genetic algorithm, showing a very rich and exhaustive representation of the information. The genotype of an individual in BIANCA can be tailored in order to represent any assembly of optimisation variables, i.e. points belonging to any design space, either homogeneous or mixed in nature.

In terms of numerical strategies, the authors propose an original technique for the handling of constraints, based on a generalisation of the classical penalisation method. Engineering optimisation problems are almost systematically subject to equality and/or inequality constraints, and in the case of multimodal non linear optimisation problems, the global constrained minimum is often located on the border of the design domain, which makes it difficult to be found. The *automatic dynamic penalisation* in BIANCA proved to be effective in solving single-objective as well as multi-objective constrained problems.

In this paper, we illustrate the architecture of the new version of the genetic algorithm BIANCA, the numerical methods that we developed, and we give a detailed description of the representation of information, i.e. the coding of populations and individuals within BIANCA. In the end, we give a number of examples of solutions found by running BIANCA on various optimisation problems, which we chose as benchmarks, and in applications to optimal design of composite laminates.

2 The architecture of BIANCA

Originally a standard genetic algorithm, BIANCA was created by Vincenti and Vannucci based on a single population evolving along several generations by means of genetic operators of one-point crossover, mutation and elitism. Selection was based on a roulette wheel method, and stop criterion was a given number of new generations produced. In this form, BIANCA allowed us to treat single-objective unconstrained optimisation problems, since originally we dealt with a single highly non convex objective function resuming all design criteria related to elastic symmetries of laminates [7–9]. In this case, the optimisation variables are all the constitutive parameters of the laminate (orientation angles, materials of the elementary layers, etc.), which can be continuous, discrete and/or grouped. Later on, we introduced some additional design criteria in the form of inequality constraints: for instance, tailoring stiffness properties could be expressed as constraints on the minimum values of Young's moduli for in-plane and bending behaviors. Therefore, a further development of BIANCA was the introduction of a technique for handling inequality constraints, based on an *evolutionary death-penalty strategy* with progressively increasing constraint level along the generations. A description of the architecture of BIANCA in its first version can be found in [7,8].

However, it is not always possible to estimate the appropriate level of constraints to introduce in an engineering design problem, especially when the real aim of the designer is to optimise such and such property. For instance, in the case of composite laminate design we find more natural to put the general problem in the form of optimizing several criteria (minimizing weight, maximizing stiffness and/or strength, etc.) under the condition of respect of elastic symmetries (uncoupling, orthotropy, isotropy, etc.). Of course, we enter here the domain of multi-objective optimisation, and we introduce difficult equality and/or inequality constraints to the optimisation problem.

Aware of the increased complexity of the problem, we re-developed BIANCA from scratch and we made it a more powerful and modular numerical tool, which we can apply also to more general problems of engineering optimisation. As far as the architecture of BIANCA is concerned, we decomposed it in the form of macros that we can assembly in various ways in order to suit the particular problem we deal with, and also to test the effectiveness of different numerical strategies. In this sense, the new version of BIANCA is not a single genetic algorithm, but a bunch of genetic tools which we can use as bricks to build up several genetic or evolutionary algorithms.

The first step in BIANCA is the initialization of the algorithm based on a given number of input values. The input information includes both problem definition and genetic parameters defining the numerical strategy that the algorithm will follow in order to solve the problem. First of all, the choice is between working on a single population or on multiple populations, whilst the problem definition (i.e. objective and constraints functions, optimisation variables) implies a given structure for individuals: we give a detailed description of genotype representation in BIANCA in the next section of this paper.

Once the initial population is created in BIANCA, each step of the algorithm can be performed according to different methods, and even several strategies can be applied at the same time. For sake of synthesis, we will give here a short overview of some relevant qualities of BIANCA.

- Objective function evaluation: a library of functions is available in BIANCA corresponding to objective functions and constraints of various optimisation problems and benchmarks.
- Fitness evaluation: several choices are available for fitness evaluation depending on the kind of problem (minimization/maximization) and on the selection pressure we decide to introduce. Here we show two basic cases of fitness evaluation in the case of minimization and maximization problems, respectively. In expressions (1) and (2), we can adjust the pressure of selection tuning the value of parameter C:

$$\text{fitness} = \left(1 + \frac{f_{\text{obj}} - \min\limits_{\text{pop}}\left(f_{\text{obj}}\right)}{\min\limits_{\text{pop}}\left(f_{\text{obj}}\right) - \max\limits_{\text{pop}}\left(f_{\text{obj}}\right)}\right)^{C}, \quad C > 1 \tag{1}$$

$$\text{fitness} = \left(1 + \frac{f_{\text{obj}} - \max\limits_{\text{pop}}\left(f_{\text{obj}}\right)}{\max\limits_{\text{pop}}\left(f_{\text{obj}}\right) - \min\limits_{\text{pop}}\left(f_{\text{obj}}\right)}\right)^{C}, \quad C > 1 \tag{2}$$

- Selection: we programmed in BIANCA several known techniques of selection (roulette wheel, tournament, ranking).
- Genetic operators: the main genetic operators in BIANCA are one-point crossover and mutation, applying with a given probability on each gene of the individual genotype (see the description of genotype in Sect. 3).

- Additional genetic operators: we also keep elitism among possible genetic operators in BIANCA, since it proved to be effective in the previous version of the algorithm. In the new version of BIANCA, we can skip elitism if we make use of some selection strategies, such as ranking.
- Handling multiple populations: the need to simultaneously explore different regions of the design space, as well as the search for optima responding to distinct design criteria, led us to introduce the option of working with multiple populations in BIANCA. We describe the structure of a population in BIANCA in Sect. 3, but we can say here that a migration operator is introduced in order to allow exchanges of information between populations evolving through parallel generations. The migration technique applied in BIANCA is ring-type, and other strategies can be introduced also.
- Stop criterion: maximum number of generations reached or test of convergence (no improvement of the mean fitness of the population after a given number of cycles).

3 Coding the genotype: the representation of the information in BIANCA

Composite design problems are very hard combinatorial optimisation problems, ruled by a high number of criteria as well as a manifold of design parameters [6,7]. As it is the case in many engineering optimisation problems, design variables for composite laminates can be of various types, but in real-world design problems they are essentially discrete. For instance, orientation angles must be chosen within a set of limited standard values in order to assure feasibility and to reduce losses of material. In a similar way, constitutive materials for the elementary layers must be chosen within a set of commercially available materials, and they represent some kind of grouped variables, since the choice of one material for one layer corresponds to the choice of all mechanical properties of the layer. But also a different approach can be imagined, where the elementary layer is itself one object of the design process, and its constitutive components (type of fibers, type of polymer matrix, volume fraction of fibers, thickness) become themselves design parameters, being also discrete or grouped variables. Moreover, in order to assure the generality of the design process the engineer has to leave the complete freedom in the definition of variability sets for all design parameters, and all design variables must be coded within the genetic model representing a composite laminate.

It is worth noting that all the considerations mentioned here related to optimal design of composite laminates apply also to more general engineering optimisation problems. Therefore, when we put a special effort in the development of the new version of BIANCA in order to solve more complex problems in composite optimal design, we actually build a very general and adaptable numeric tool to solve various problems of structural optimisation.

It appears clearly that one of the main issues in this sort of problems is the representation of the information, which has to be detailed and exhaustive, but also non redundant.

The biological metaphor in genetic algorithms is a simple but powerful means to restitute the richness and completeness of information linked to design variables. In fact, the concept of coding the characteristics of every composite laminate in a genotype leaves the freedom to enrich the genotype structure as much as necessary (and as much as possible).

The necessity to deal with discrete and grouped variables leads us to the choice of a discrete representation of the information. As already programmed in the previous version of our algorithm, discrete variables are represented by integer numbers, which are pointers referencing to the set of feasible discrete values for each variable (see Sect. 3.1). In a standard genetic algorithm, it is usual to encode integer values in the form of binary strings, in order to use the minimalist alphabet which increases the number of schemes. According to literature on genetic algorithms, that assures improvement of the exploitation of information and the exploration of the optimisation search space [4,5].

Additionally, the binary representation of variables allows the use of binary crossover and mutation which proved to be effective when dealing with particular classes of engineering optimisation problems, such as the optimal design of laminates. These problems are ruled by severely non linear and non convex objective functions showing steep variations along very short distances within the design space [7–9]. Therefore, the result of binary crossover, leading to offsprings possibly located far away from their parents, is not necessarily a disadvantage, and it assures a large exploration of the search space.

In the following sub-sections, we describe our methodology to encode/decode values of variables based on a binary alphabet (Sect. 3.1), our operations of crossover and mutation using Boolean operators (Sect. 3.2) and we illustrate the structure representing the genotype of individuals in BIANCA (Sect. 3.3).

3.1 Encoding/decoding values of variables in BIANCA

Our methodology is based on a representation of the domain of definition for each variable in the optimisation problem by the use of pointers, which are themselves integer values. If dealing with discrete or grouped variables with domain of definition of finite dimension N, we can enumerate all admissible values v_i $(i = 1,\ldots,N)$ and build a reference between each value v_i and its index i, which is therefore the pointer to the value v_i. In case the domain of definition of a variable is not of finite dimension, it is necessary to restrict it and to define an upper bound $v_{\max}$ and a lower bound $v_{\min}$ to the space of admissible values v_i.

In the case of continuous real variables, we proceed to the discretisation of the domain of definition by choosing a given precision p (the precision parameter can take different values for each continuous variable), in order to apply the same system of referencing by pointers as for discrete and grouped variables (Fig. 1).

In BIANCA, pointers constitute the genotype of an individual (precisely, a pointer corresponds to a gene, as illustrated in Sect. 3.3), and genetic operators of crossover and mutation directly apply on the pointers representing each variable. Therefore, a step of decoding/encoding is necessary to translate the value of the pointer into the corresponding value of the variable, and viceversa; but this is not time consuming, since the table of reference between the admissible values for each variable and the corresponding pointers is built once for all at the launch of a run of BIANCA. Since pointers are integer values, no binary encoding/decoding is necessary, and genetic operators are expressed as proper combinations of Boolean operations over integer numbers (see Sect. 3.2). This methodology assures that all variables are treated and manipulated in the same way and at the same time within our genetic algorithm.

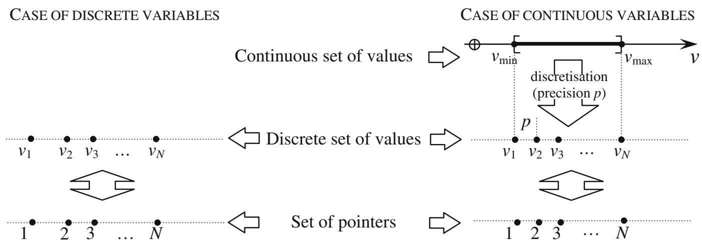


Fig. 1 Correspondence between values of variables and pointers within BIANCA

3.2 Cross-over and mutation using Boolean operators

We illustrate here how we perform the operations of crossover and mutation directly over the integer values of pointers by the use of Boolean operators. Our approach is based on the computer-embedded representation of integers as sequences of binary digits. Being N the number of pointers for a given variable (see Sect. 3.1), let l be the length (number of digits) of the corresponding binary sequence (so that: $2^{l-1} \leq N \leq 2^l - 1$).

One-point crossover between two corresponding genes of two individuals (parents), represented by pointers I_1 and I_2, is performed in two steps: after the generation of a random position m ($1 \leq m \leq l-1$) for the recombination of genes, the first step is the extraction of the left and right segments of each parent gene, I_i^{left} and I_i^{right} ($i = 1, 2$). For this purpose, we build two auxiliary numbers, $I_{\text{aux}}^{\text{left}}$ and $I_{\text{aux}}^{\text{right}}$:

$$\begin{aligned} I_{\text{aux}}^{\text{left}} &= 2^l - 2^m \\ I_{\text{aux}}^{\text{right}} &= 2^m - 1 \end{aligned} \tag{3}$$

Then, we can simply generate the left and right segment for each parent gene using a Boolean "AND" operator:

$$\begin{aligned} I_i^{\text{left}} &= I_i \quad \text{AND} \quad I_{\text{aux}}^{\text{left}} \\ I_i^{\text{right}} &= I_i \quad \text{AND} \quad I_{\text{aux}}^{\text{right}} \end{aligned} \quad (i = 1,\ 2) \tag{4}$$

The second step for crossover is the recombination of the segments extracted from the parent genes in order to produce two children genes, C_1 and C_2. This is performed using a Boolean "OR" operator:

$$\begin{aligned} C_1 &= I_1^{\text{left}} \quad \text{OR} \quad I_2^{\text{right}} \\ C_2 &= I_2^{\text{left}} \quad \text{OR} \quad I_1^{\text{right}} \end{aligned} \tag{5}$$

In the same way, we perform mutation over a gene I on one of his bits at a random position m. Firstly we generate an auxiliary mutation number $I_{\text{aux}}^{\text{mutation}}$, defined as:

$$I_{\text{aux}}^{\text{mutation}} = 2^{m-1} \tag{6}$$

We can use $I_{\text{aux}}^{\text{mutation}}$ to check the binary digit at position m whether be 0 or 1 through the generation of a number, that we call *chk*:

$$chk = I \text{ AND } I_{\text{aux}}^{\text{mutation}} \tag{7}$$

Finally, the mutated individual I_{mut} will be determined according to the rule (8):

$$\begin{aligned} &\text{if } chk = 0 \text{ then:} \\ &\quad I_{\text{mut}} = I \quad \text{OR} \quad I_{\text{aux}}^{\text{mutation}} \\ &\text{else:} \\ &\quad I_{\text{mut}} = I \ - \ I_{\text{aux}}^{\text{mutation}} \end{aligned} \tag{8}$$

Finally, Boolean operations can be arranged also in order to treat multiple-point or uniform crossover if necessary, but this is not applied in BIANCA at the moment.

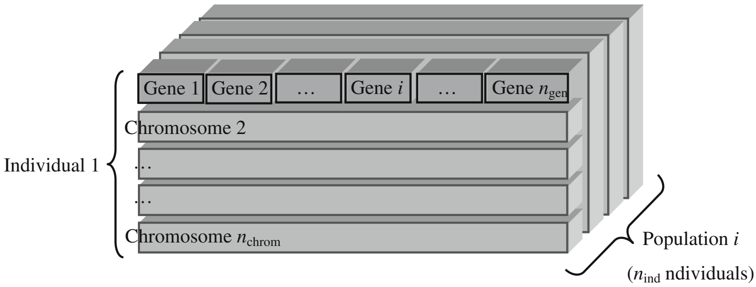


Fig. 2 Structure of a population in BIANCA

3.3 The structure of the individual genotype in BIANCA

Concerning the structure of the genotype in BIANCA, we already introduced the concept of general and exhaustive representation of the genetic code of individuals in the first version of the algorithm, but our effort was focused on the optimisation of laminate stacking sequences and we limited the scope to the basic parameters for laminate design, i.e. orientation angles [7,8]. Nevertheless, we have to take into account additional constitutive parameters when designing composite laminates, such as constitutive materials, number of layers, and so on. This is the reason why in the present version of BIANCA we improve the structure of the genotype of individuals.

An individual in BIANCA is an array of n_{chrom} chromosomes, each chromosome being an array of n_{gene} genes (Fig. 2). Basically, each design variable is coded in the form of a gene, and its meaning is linked to the gene position within the chromosome. No limit is imposed on the number of genes and chromosomes for an individual in BIANCA, and we can also imagine individuals composed of chromosomes of various lengths, i.e. made of different numbers of genes. A number n_{ind} of individuals compose a population, and we can have several distinct populations evolving in BIANCA. The whole set of information is therefore included in a four-dimension array, which we call *being*$(n_{\text{pop}}, n_{\text{ind}}, n_{\text{chrom}}, n_{\text{gene}})$.

As an example, a composite laminate is represented within BIANCA as an assembly of chromosomes, one for each layer of the laminate; and each chromosome is composed of several genes representing the layer constitutive parameters: we will have a gene for the orientation, one for the constitutive material, one for the layer thickness, and so on.

In any case, crossover is performed systematically on each corresponding couple of genes belonging to two individuals (parents) in order to deeply mix genetic material between individuals.

We can remark that the representation of populations and individuals in BIANCA is very general, which opens the way to the application of this genetic algorithm to various problems in engineering optimisation.

4 Handling of constraints in BIANCA: evolutionary death penalty and automatic dynamic penalisation

Several authors put an effort in the development of appropriate and effective strategies for genetic algorithms in order to deal with constrained optimisation problems [10–13]. We intended to give our contribution in this domain and we focused our interest on the difficult case of the constrained global minimum located on the frontier of the design space. To this extent, we developed a strategy which is based on the combination between classical penalisation methods and the exploitation of the distributed information over the population in a genetic algorithm. We called our approach the *automatic dynamic penalisation*.

In the original version of BIANCA, the authors developed a method to handle optimisation problems with inequality constraints, called *evolutionary death penalty*. This strategy was a combination of progressive increase of the constraint level together with death penalty: that is to say, individuals which do not respect the active level of constraints are eliminated from the current population [8].

The idea is that, in the first phase of the solving process, the constraint level is kept lower than the final required value, in order to largely and exhaustively explore the search space. As the optimisation process gets closer to better individuals in terms of objective function, the level of constraint is raised and that restricts the area of the search to the current feasible area, which on its turn gets to coincide with the final feasibility domain for the optimisation problem. The initial value and the increasing rate of the constraint level are set as numeric parameters in the genetic algorithm, and they can be tuned by the user.

This strategy proved to be effective in application to optimal design problems of laminates where the requirements are some kind of elastic symmetries for the laminates with constraints on their stiffness response [8]. Nevertheless, it might be difficult to properly set the numeric parameters ruling this strategy. And mainly, the drastic elimination of unfeasible individuals from the search process corresponds to a loss of points within the whole search space (feasible and unfeasible domains), and in some sense it is a spoil of precious genetic information. Moreover, the risk is that the population of the genetic algorithm might quickly converge towards an optimal or quasi-optimal point outside the final feasible domain, and therefore an unacceptable solution for the optimisation problem.

For these reasons, we keep the possibility to apply the *evolutionary death penalty* strategy in BIANCA, but we also developed a more evolved strategy for handling of constraints in the new version of the algorithm, the *automatic dynamic penalisation*.

Given a standard optimisation problem with an objective function $f(\mathbf{x})$ and n_g inequality constraints $g_k(\mathbf{x})$, being $\mathbf{x}$ the vector of optimisation variables:

$$\begin{aligned} &\min\ f(\mathbf{x}) \\ &\text{such that}: g_k\ (\mathbf{x}) \leq 0\ ,\ k = 1, \ldots, q \end{aligned} \tag{9}$$

classical penalisation methods [1] transform it into a non-constrained problem through the definition of a new modified objective function $F(\mathbf{x})$:

$$\begin{gathered} \min\ F(\mathbf{x}) \\ \text{where}: F(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & \text{if}: g_k(\mathbf{x}) \leq 0, \quad k = 1, \ldots, q \\ f(\mathbf{x}) + \sum_k c_k g_k(\mathbf{x}) & \text{if}: g_k(\mathbf{x}) > 0, \quad k = 1, \ldots, q \end{cases} \end{gathered} \tag{10}$$

In expression (10), parameters c_k are penalisation coefficients and the user must set their values to an appropriate level in order to assure the search of solutions for the optimisation problem to be forced within the feasible domain (points respecting the constraints). Nevertheless, the choice of coefficients c_k is very difficult, and it is common use to estimate their values by trial and error. Moreover, it could be useful and effective to adjust penalisation pressure along the optimisation process by tuning the penalisation coefficient, but again this can only rely on a guess or on a deep knowledge of the nature of the optimisation problem by the user.

Since genetic algorithms perform the evolution towards the global optimum of a population of individuals scattered over the search space, we can exploit this distributed information in order to guide the search in the case of constrained optimisation problems. Generally, in the first step of the algorithm, the randomly generated population is evenly distributed over the feasible and unfeasible domain, and the corresponding values of objective function and constraints can be used to estimate an appropriate level of penalisation, i.e. values of penalisation coefficients c_k.

In fact, in a population we can separate feasible and unfeasible individuals and we can classify each group in terms of increasing values of their objective function. The first individual in each group is the best candidate to be solution of our problem on the feasible and unfeasible side, respectively. We call f_0^F and f_0^{NF}, g_0^{NF} the values of the objective function and of constraint for these two individuals (apex F stands for *Feasible*, and *NF* for *Non Feasible*). The penalisation coefficient is estimated so that the best unfeasible individual has at least the same value of the modified objective function $F(\mathbf{x})$ as the best feasible individual, that is to say:

$$c = \frac{\left| f_0^F - f_0^{NF} \right|}{g_0^{NF}} \tag{11}$$

Of course, the estimation of the penalisation coefficient according to expression (11) can be repeated at each generation, thus tuning the appropriate penalisation pressure on the current population.

We name our strategy as the *automatic dynamic penalisation method* in the genetic algorithm BIANCA: *automatic* because the algorithm can automatically calculate the penalisation coefficients c_k, and *dynamic* since the evaluation of the penalisation level is updated at each iteration, i.e. penalisation coefficient are actualised at each generation in a run of BIANCA.

The idea is that the genetic search should lead towards feasible global optima, thus penalising all the unfeasible individuals. Nevertheless, we know that often optimal points in constrained optimisation problems can be located on the border of the feasible and unfeasible areas, and the case is also possible of such minima being close to points belonging to the unfeasible domain but showing an optimal value of the objective function (see our benchmark problem in Sect. 6). Our method allocates a good evaluation to individuals belonging to the unfeasible domain but showing a very good value of the objective function, and therefore it allows to reach the constrained optimum by leading the search through the unfeasible domain, i.e. across the shortest and most effective path.

We tested successfully the automatic dynamic penalisation method on a benchmark problem (see Sect. 6.1), as well as on constrained optimisation problems for composite laminates (see Sect. 7).

5 Multiobjective constrained optimisation in BIANCA

Often in real world engineering problems, designers have to deal with multi-objective optimisation problems and, most of times, distinct objectives can be conflicting [1–3]. A classical strategy to deal with multi-objective optimisation is to build the Pareto front for a given problem, that is to say the set of the non-dominated solutions, and eventually the choice is left to the designer to pick any point of the front as a possible solution. Some authors tried to develop strategies based on the game theory in order to automatically guide the optimisation algorithm towards the choice of an optimal solution [14,15].

Since in a first time we intended to deal with multi-objective constrained optimisation problems for composite laminates, we decided to introduce strategies in BIANCA which allowed us to treat this class of problems through the search of the Pareto front and we coupled it with our method for handling of constraint.

One first strategy for the construction of the Pareto front is based on the classical idea of the weighted sums [1–3,14,16]. The idea is to launch several runs of the genetic algorithm BIANCA over mono-objective optimisation problems, where the auxiliary objective function $F(x)$ is defined as a combination of the current objective functions $f_i(x)$, in the form of a weighted sum. Weight coefficients ω_i vary for each launch, so that they respect the following conditions:

$$0 \leq \omega_i \leq 1 \quad \text{and} \quad \sum_i \omega_i = 1 \tag{12}$$

By the variation of the weight coefficient over their definition domain, the algorithm is able to build the Pareto front point by point. Nevertheless, the result is an incomplete and discontinuous Pareto front [17], according to [3,14–16].

In order to improve the results on the construction of the Pareto front, we chose to apply a combination of the fitness sharing strategy together with the niche method, as it is described in [14–16]. The idea is to guide the genetic algorithm to converge towards a final population of non-dominated individuals, representing the Pareto front for the current multi-objective optimisation problem.

Therefore, the strategy is, at each generation of the genetic algorithm, to rank the individuals within a population on the basis of their dominance, in order to identify a number of fronts of increasing degree of dominance. If an individual i is dominated by n_{dom} individuals within the current population, then its rank r_i is:

$$r_i = 1 + n_{\text{dom}} \tag{13}$$

and individual i shares the same rank as other individuals belonging to the same front (i.e. having the same dominance). The fitness value f_i for each individual is calculated as:

$$f_i = \frac{1}{r_i} \tag{14}$$

Yet, this value of the fitness is modified on the basis of the neighbouring of each individual: the idea is to degrade the fitness of an individual when it has a large number of close neighbours sharing the same dominance, in order to avoid the genetic search to converge on few isolated points within each front.

A group of neighbours within the same front, which are closer than a given distance σ_{share}, constitute a niche. For each couple of individuals (i, j) in a niche, a sharing function $Sh(d_{ij})$ is defined:

$$Sh(d_{ij}) = \begin{cases} 1 - \frac{d_{ij}}{\sigma_{\text{share}}} & \text{if} \quad d_{ij} < \sigma_{\text{share}} \\ 0 & \text{if} \quad d_{ij} \geq \sigma_{\text{share}} \end{cases} \tag{15}$$

where d_{ij} is the distance between two individuals and σ_{share} is the threshold distance for two individuals to be members of the same niche. The density m_i of the niche around a single individual i is therefore evaluated as:

$$m_i = \sum_j Sh(d_{ij}) \tag{16}$$

Finally, the fitness f_i is modified to $\tilde{f_i}$ on the basis of the niche density:

$$\tilde{f_i} = \frac{f_i}{m_i} \tag{17}$$

This strategy results in a population at the last generation which contains the Pareto front for the considered problem.

We couple this strategy with the *automatic dynamic penalisation* method for constraint handling, and we will show that this approach succeeds in building continuous and compact Pareto fronts for problems of multi-objective constrained optimisation (see Sects. 6.2 and 7.4).

6 Applications of BIANCA on benchmark problems

In this section, we show results obtained by running BIANCA over a few problems of mathematical optimisation that we chose as benchmarks for our genetic algorithm. Our aim was especially to test new strategies developed in BIANCA in order to deal with constrained problems and multi-objective optimisation. Therefore, we show results on a benchmark for each class of problems in the following Sects. 6.1 and 6.2. More tests on benchmark functions can be found in [17].

6.1 A constrained optimisation problem: Vannucci's problem

The constrained optimisation problem proposed by Vannucci is defined as:

$$
\begin{aligned}
&\min f(x) = -e^{ka\sqrt{x_1+x_2}} \sin(ax_1)\cos(2bx_2) \\
&\text{subject to:} \\
&\quad x_2 > e^{cx_1^2} - 1 \\
&\quad 0 \leq x_1 \leq 4\pi \\
&\quad 0 \leq x_2 \leq 2\pi
\end{aligned}
\tag{18}
$$

Parameters k, a, b and c can be chosen in order to change the shape of the objective and constraint functions.

In Fig. 3, we represent the functions defining the problem (18) in case: $k = 0.2$, $a = 1$, $b = 0.6$, $c = 0.012$. We can notice that this problem is difficult because of the high non convexity of the objective function; moreover, the global optimum is located on the intersection between the objective and constraint function, that is to say on the border of the feasible domain, and very close to the unconstrained global optimum, which plays the role of a strong attraction point for the genetic search.

The population size in BIANCA was set to $n_{\text{ind}} = 120$, and the maximum number of generations to $n_{\text{gen}} = 200$. The probabilities for cross-over and mutation are $p_{\text{cross}} = 0.82$ and $p_{\text{mut}} = 0.02$. Genotypes are coded according to the pointer method, selection is performed by tournament between two individuals, and elitism is active ($n_{\text{elite}} = 1$).

We applied the *classical static penalisation* method (i.e. the designer evaluates or guesses an appropriate value for the penalisation coefficient c, which stays fixed along the iterations), as well as the *automatic static* (i.e. the algorithm automatically calculates the penalisation coefficient, as explained in Sect. 4, only at the first generation) and *automatic dynamic penalisation* methods (i.e. the algorithm automatically calculates the penalisation coefficient, as explained in Sect. 4, and updates it at each generation).

The aim is to compare the robustness and precision of these methods, and we elaborated different parameters of reliability [17]. We describe here the most significant, which is the percentage of runs of BIANCA giving the constrained global optimum as a final solution. In Fig. 4, we illustrate the results of reliability for the three methods evaluated on several

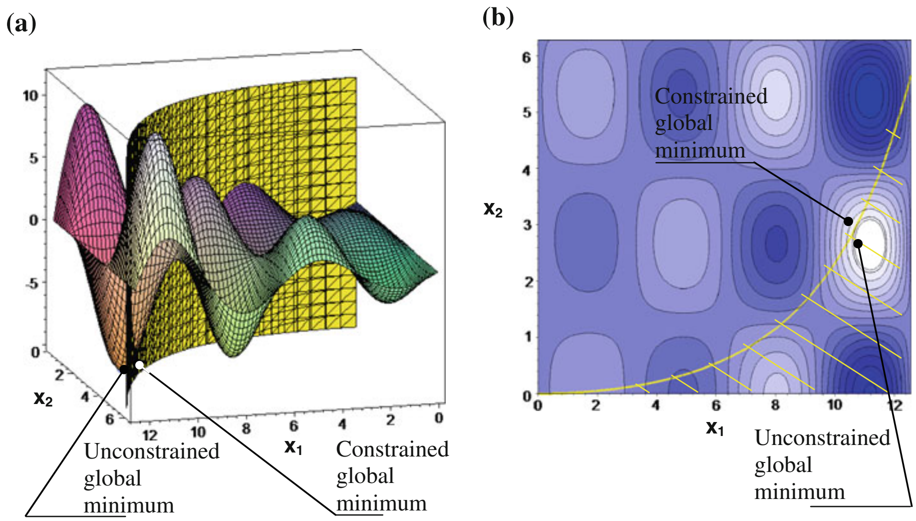


Fig. 3 Vannucci's problem: objective and constraint functions. **a** Three-dimensional view. **b** Curves of level

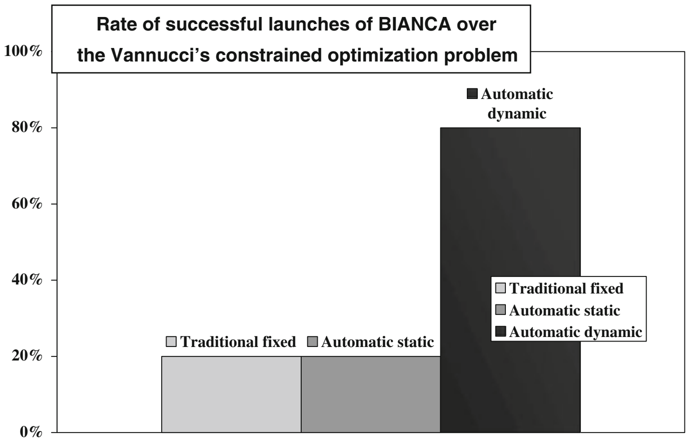


Fig. 4 Comparison of reliability of different penalisation strategies: classical static penalisation, automatic static and automatic dynamic

launches of the algorithm BIANCA on the Vannucci's problem. We can remark that the value of reliability calculated for the traditional fixed penalisation method is the same as the automatic static one; yet, we can remind that in the traditional method, several attempts might be necessary in order to estimate the appropriate level of the penalisation coefficient. The value of reliability dramatically increases when using the *automatic dynamic penalisation*.

We give the best result found by BIANCA in Table 1 together with the exact constrained optimum point. Even if the theoretical optimum point is known a priori, this information is not exploited when running the genetic algorithm.

In Fig. 5, we illustrate the evolution along the generations in BIANCA of the percentage of feasible individuals over a population (Fig. 5a) and the average violation of constraint (Fig. 5b).

Table 1 The exact solution for problem (18) and numerical solution by BIANCA

	Exact solution	Solution by BIANCA
Point of global minimum	(10.712; 2.693)	(10.712; 2.692)
Value of global minimum	−8.116	−8.116

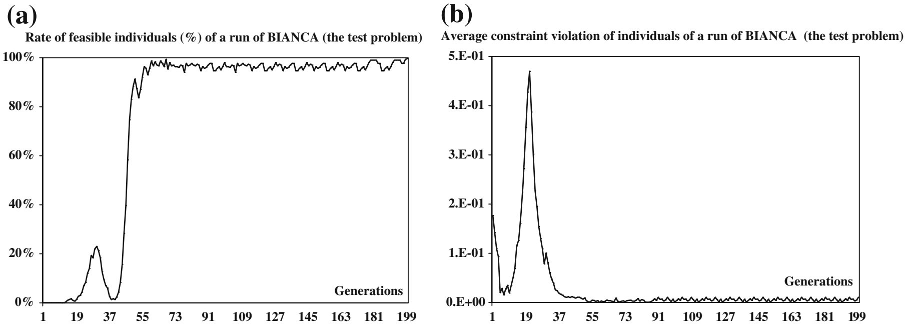


Fig. 5 **a** Rate of feasible individuals and **b** average constraint violation along generations in BIANCA with automatic dynamic penalisation

In conclusion, we can say that the automatic dynamic penalisation method proved to be a powerful strategy especially in the case of Vannucci's problem, where the global minimum is situated on the border between feasible and unfeasible areas; problems of this kind are known as very hard to solve optimisation problems.

6.2 Application to multi-objective optimisation problems

We consider two multi-objective optimisation problems as benchmarks, chosen from the literature [16], which we treated by the "ranking and niche" method for the construction of the Pareto front described in Sect. 5.

The first benchmark problem consists of two objective functions to minimize:

$$\begin{array}{ll} \min & \begin{array}{l} f_1(x) = (x_1 - 1)^2 + (x_2 - 3)^2 \\ f_2(x) = (x_1 - 4)^2 + (x_2 - 2)^2 \end{array} \\ \text{subject to:} & \begin{array}{l} -5 \le x_1 \le 5 \\ -5 \le x_2 \le 5 \end{array} \end{array} \tag{19}$$

Using the non-dominated ranking method (see Sect. 5) over a population of $n_{\text{ind}} = 50$ and $n_{\text{gen}} = 50$ generations, the genetic algorithm BIANCA gave the Pareto front for problem (19), as illustrated in Fig. 6. We can notice that the resulting Pareto front is rather continuous and well built.

A more complex example is problem (20), where we have three objective functions [16]:

$$\begin{array}{ll} \min & \begin{array}{l} f_1(x) = x_1^2 + (x_2 - 1)^2 \\ f_2(x) = x_1^2 + (x_2 + 1)^2 + 1 \\ f_3(x) = (x_1 - 1)^2 + x_2^2 + 2 \end{array} \\ \text{subject to:} & \begin{array}{l} -2 \le x_1 \le 2 \\ -2 \le x_2 \le 2 \end{array} \end{array} \tag{20}$$

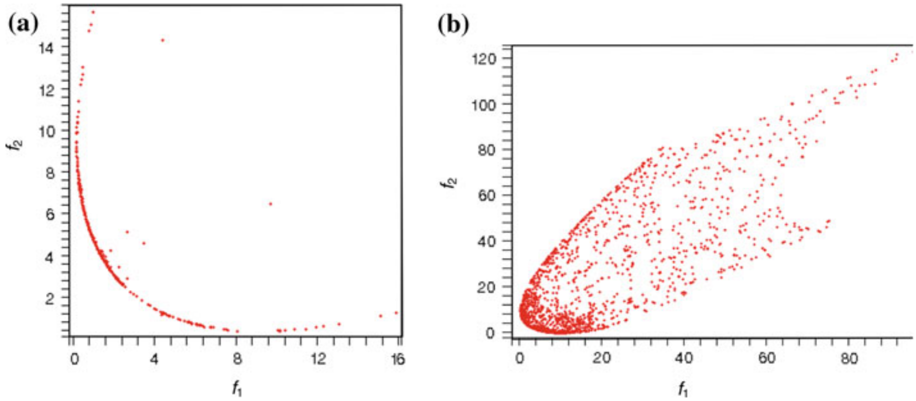
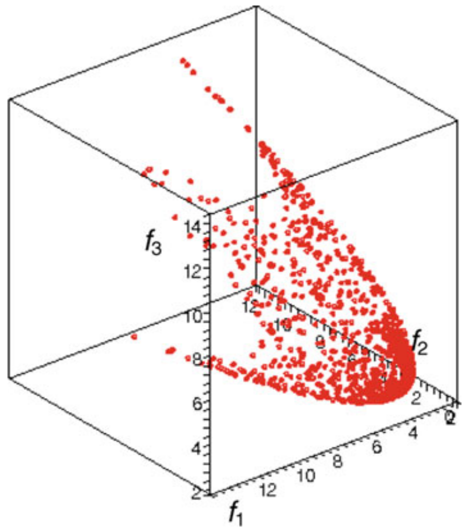


Fig. 6 **a** Pareto front for problem (19) at the 50th generation of BIANCA and **b** distribution of individuals in five different generations of BIANCA

Fig. 7 The 1st, 25th and 50th generations of BIANCA and the three-dimensional Pareto front for problem (20)



Using the non-dominated ranking method (see Sect. 5) over a population of $n_{\text{ind}} = 100$ and $n_{\text{gen}} = 50$ generations, the genetic algorithm BIANCA gave the Pareto front for problem (20), as illustrated in Fig. 7.

7 Applications of BIANCA to real-world engineering problems: optimal design of composite laminates

The first version of the genetic algorithm BIANCA was specifically developed in order to solve design problems for composite laminates, which we formulated as an unconstrained single-objective optimisation problem [7–9]. Our further works on a global formulation for the optimal design of composite structures [17,18] urged us to extend and enrich BIANCA in order to treat much more complex optimisation problems (constrained problems with equality and/or inequality constraints; multi-objective problems; etc.). Our effort in the development of new numerical strategies within BIANCA was driven by the reflection on more general applications to optimisation problems in engineering, so that the structure of BIANCA is very general and versatile. Nevertheless, our first application of the new version of BIANCA to real-world engineering problems concerned the optimal design of composite laminates and structures, which we illustrate in this section. After a brief introduction to our formulation of the optimisation problems (Sect. 7.1), we will show a few examples of results found using BIANCA (Sects. 7.2–7.4).

7.1 Formulation of optimal design problems for composite laminates

A composite laminate is a stack of layers oriented at various angles, each layer made of a fiber-reinforced matrix (glass, carbone or aramid fibers embedded in a polymer matrix is the most common composition of current structural composites). They are usually employed in the construction of plates and shell-like structures in many domains (aeronautics, automotive, naval, sports, biomedical, energy production, etc.). On one hand, because of their heterogeneous architecture, composite laminates show peculiar behaviours when compared to more traditional structural materials, such as metals; mainly, they are generally anisotropic and have several couplings among different behaviours (for instance, an elastic coupling between the in- and out-of-plane behaviours). On the other hand, the very structured nature of such materials allows the designer to tailor a laminate in order to match some required properties.

Many authors have dealt with the optimal design of composite laminated plates in terms of their constitutive parameters (number of elementary layers, materials and orientations of the elementary layers) and with respect to various properties (stiffness, strength, buckling resistance, vibration, weight, etc.) (see [6,19] and references therein). Nevertheless, the complexity of behaviours of such materials (particularly, anisotropy and couplings) classically induces authors to the introduction of some simplifying hypothesis in the formulation of the design and/or optimisation problems, so that the search space is drastically reduced. For instance, it is generally accepted to limit the search to symmetrically stacked laminates in order to assure elastic uncoupling.

Our contribution specifically concerns the development of a general and global approach to the design of composite laminated plates, where all the required properties are explicitly expressed as criteria of the optimisation problem, either in the form of objectives or of constraints. Within all the considered properties, we also include all elastic symmetries and (un)couplings, which are always active as objectives or constraints in our formulation.

The general expression of our formulation is:

$$\begin{aligned} &\min\ f(\mathbf{x}) \\ &\text{such that: } \begin{cases} g_k(\mathbf{x}) \leq 0,\ k = 1, \ldots, q \\ I(\mathbf{P}(\mathbf{x})) < \varepsilon \\ \mathbf{x}_L \leq \mathbf{x} \leq \mathbf{x}_U \end{cases} \end{aligned} \tag{21}$$

where $f(\mathbf{x})$ is the objective function; $g_k(\mathbf{x})(k = 1, \ldots, q)$ are constraint functions; $I(\mathbf{P}(\mathbf{x}))$ is the constraint over the respect of given elastic symmetries [7,9,17,18] (ε is a precision parameters); finally, we can have box constraints over the design variables $\mathbf{x}$. Vector $\mathbf{x}$ represents all the constitutive parameters of a composite laminate (number of elementary layers, materials and orientations of the elementary layers). It is worth noting that the optimisation variables can be continuous, but they are essentially discrete and/or grouped.

In the following sub-sections, we will illustrate the details of formulation (21) in its applications to the optimal design of laminates with respect to various properties (two cases of single-objective problems, and a multi-objective constrained one) using the new version of BIANCA.

7.2 Maximisation of buckling load for composite laminated plates

Instability of laminated plates is a very important issue when these very thin structures are loaded in compression. We consider here the case of a rectangular simply supported uncoupled orthotropic laminated plate loaded in compression by in-plane loads of stress resultants N_x and N_y, whilst shear load N_{xy} is zero (Fig. 8); a and b are the lengths of the plate sides.

In this case, buckling shapes are sinusoidal, and being m and n the number of half waves in the x and y direction respectively, the critical value of the load multiplier $\lambda_{\text{crit}}(m,n)$ inducing buckling is:

$$\lambda_{\text{crit}}\,(m,n) = \frac{\pi^2\left[D_{11}(m/a)^4 + 2(D_{12}+2D_{66})(m/a)^2(n/b)^2 + D_{22}(n/b)\right]^4}{(m/a)^2\,N_x + (n/b)^2\,N_y} \tag{22}$$

In Eq. 22, $D_{ij}\,(i,j=1,\ 2,\ 6)$ are the Cartesian components of the bending stiffness tensor expressed in the reference system Oxy. We notice that expression (22) applies to orthotropic uncoupled plates, and therefore bending orthotropy is a necessary condition as well as elastic uncoupling. Moreover, the principal axes of orthotropy have to be coincident with the plate axes.

Generally, we can particularise formulation (21) for the optimisation of buckling load of a simply supported laminated plate as:

$$\begin{aligned} &\max\,(\min_{m,n}\lambda_{\text{crit}}) \\ &\text{such that} : I\,(\mathbf{P}(\mathbf{x})) < \varepsilon \end{aligned} \tag{23}$$

where function $I(\mathbf{P})$ corresponds to the necessary conditions of elastic uncoupling and bending orthotropy with the principal axes oriented along the plate axes.

We give here an example of calculation where the aspect ratio of the rectangular plate is fixed: $a/b = 1.5$, and compressive loads along the sides of the plate are: $N_x = N_y = 1$ N/mm.

We fixed the number of layers $n = 16$, whilst the optimisation variables are the stacking sequence (which is completely free) and the corresponding values of the orientation angles. The angles can take all values between $-90°$ and $90°$, with a discrete precision of $p = 1°$.

In addition, we search for a highly stiff orthotropic plate; thus, constraints are imposed over the extension Young moduli along the orthotropic axes, which are in competition with the maximisation of the buckling factor, and formulation (23) becomes:

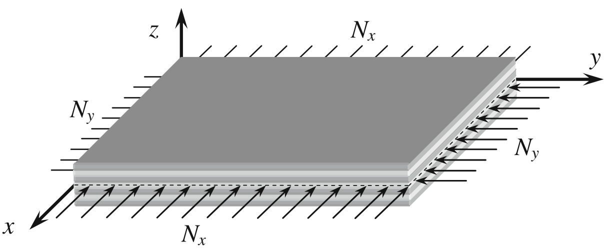


Fig. 8 A rectangular laminated plate under in-plane loads N_x and N_y

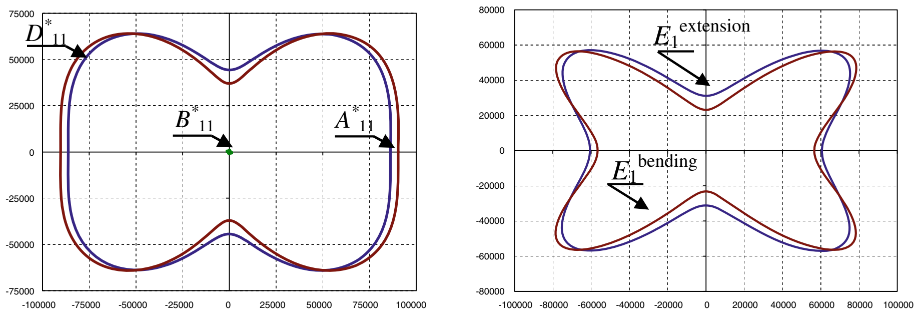


Fig. 9 Polar representation of the elastic properties for laminate (25)

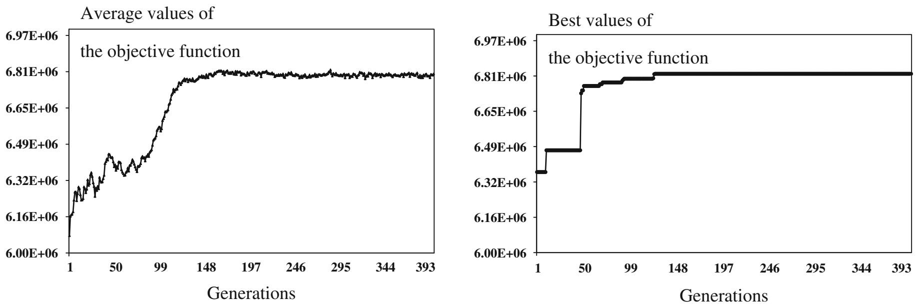


Fig. 10 Average and best values of the objective function vs. generations

$$
\begin{aligned}
&\max\ (\min_{m,n}\ \lambda_{\text{crit}}) \\
&\text{such that}: I(\mathbf{P}(\mathbf{x})) < \varepsilon \quad (\varepsilon = 10^{-4}) \\
&\qquad E_x^A \geq 60\ \text{GPa} \\
&\qquad E_y^A \geq\ 30\ \text{GPa}
\end{aligned}
\tag{24}
$$

The required elastic symmetries here, expressed by the function $I(\mathbf{P}(\mathbf{x}))$, are in-plane and bending orthotropy ($K_A = K_D = 1$), coincidence of the orthotropic axes in extension and bending, uncoupling. In addition, the orthotropic axes of the laminate must coincide with the axes of the plate. The precision of the solution in terms of elastic symmetries is fixed: $\varepsilon = 10^{-4}$.

We show here an example of solution to problem (24) found by running BIANCA using the *automatic dynamic penalisation* method:

$$[-24/39/-47/37/32/-47/-6/-47/55/59/18/-38/-38/19/-40/42] \tag{25}$$

The maximum value achieved for the buckling critical multiplier is $\lambda_{\text{opt}} = 6.86 \cdot 10^6$, whilst the achieved values for the constraint functions are: $E_x^A = 60607$ MPa, $E_y^A = 31157$ MPa and $I(\mathbf{P}(\mathbf{x})) = 8.80 \cdot 10^{-5}$. The respect of elastic symmetries is confirmed by the graphics of polar variation of stiffness properties for laminate (25), shown in Fig. 9. Units are in MPa and the horizontal and vertical axes represent the x- and y-directions, respectively.

The effectiveness of the genetic algorithm BIANCA is proved by the graphics of average and best values of the objective function over the generations, shown in Fig. 10. Calculations were run over a population composed of $n = 400$ individuals, and quasi-optimal individuals were obtained after 150 generations.

7.3 Maximisation of strength of a composite laminated plate

Maximisation of strength of composite laminates is a very delicate issue in order to prevent failure of composite plates and many authors dealt with this subject [6]. Mechanisms of failure for composite laminates are complex and related to different phenomena because of the heterogeneous nature of such materials. However, the resistance to failure of a unidirectional composite is usually measured by comparing homogeneous functions of the stresses and strains to material strength limits, and various criteria exist depending on the choice of such functions. Thus, a criterion is applied to verify the resistance to failure of each ply of a laminate (see [6] and references therein).

The need to determine a global criterion for the evaluation of the strength of a composite laminate drove us to a combined approach, based on a measure of the global strain of the laminate coupled with the verification of a local criterion for each layer of the laminate: our approach was inspired by the works of Park [19]. We used a quadratic form $f(\boldsymbol{\varepsilon})$ of the strain components to be minimised, which is a measure of the norm of the strain vector:

$$f(\boldsymbol{\varepsilon}) = \varepsilon_{xx}^2 + \varepsilon_{yy}^2 + \frac{1}{2}\gamma_{xy}^2 \tag{26}$$

Function $f(\boldsymbol{\varepsilon})$ represents the global response of the laminate to the applied state of load, and the optimisation of strength for the laminate can be written as the maximisation of function R_{index}:

$$R_{\text{index}} = \frac{1}{f(\boldsymbol{\varepsilon})} \tag{27}$$

Moreover, we chose the Hoffman criterion for the layerwise verification; for a plane state of stress expressed in the principal orthotropy directions x_1 and x_2, and for a transversely isotropic material in the plane x_2x_3, the failure envelop for the Hoffman criterion is expressed as:

$$-\frac{\sigma_{11}^2}{X_cX_t} + \frac{\sigma_{11}\sigma_{22}}{X_cX_t} - \frac{\sigma_{22}^2}{Y_cY_t} + \frac{X_c + X_t}{X_cX_t}\sigma_{11} + \frac{Y_c + Y_t}{Y_cY_t}\sigma_{22} + \frac{\tau_{12}^2}{S_{12}^2} = 1 \tag{28}$$

where X_c, X_t, Y_c, Y_t and S_{12} are the layer strength limits in tension and compression along the two material axes and for shear, respectively.

Finally, we can formulate the problem of maximisation of the strength for a composite laminated plate (which is again a particularisation of formulation (21)):

$$\begin{aligned} &\max R_{\text{index}} \\ &\text{such that:}\ f_{\text{Hoffman}} < 1 \\ &\qquad\qquad\ I(\mathbf{P}(\mathbf{x})) < \varepsilon \quad (\varepsilon = 10^{-4}) \\ &\qquad\qquad\ E_y^A \geq 45 \text{ GPa} \end{aligned} \tag{29}$$

where we added a constraint over the transverse stiffness of the plate.

We considered a rectangular laminated plate as in Fig. 8, subject to a single compressive load $N_x = 10^5$ N/mm ($N_y = N_{xy} = 0$ N/mm).

We fixed the number of layers $n = 16$, whilst the optimisation variables are the stacking sequence (which is completely free) and the corresponding values of the orientation angles. The angles can take all values between −90° and 90°, with a discrete precision of $p = 1°$.

We show here an example of solution to problem (29) found by running BIANCA using the automatic dynamic penalisation method:

$$[0/-6/-84/-5/42/4/-1/5/-72/-22/65/-84/5/-14/5/4] \tag{30}$$

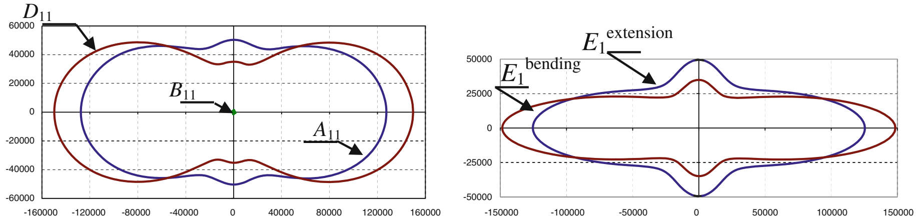


Fig. 11 Polar representation of the elastic properties for laminate (30)

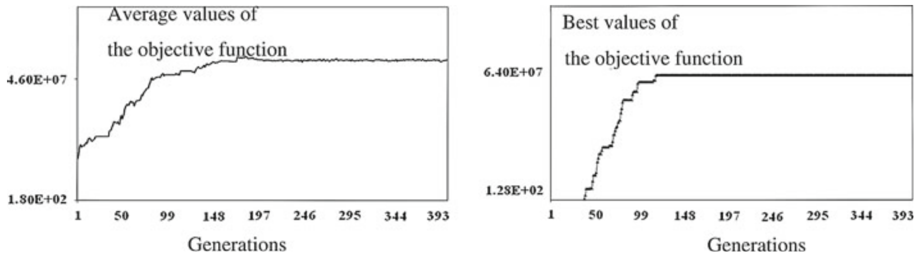


Fig. 12 Average and best values of the objective function vs. generations

The maximum value achieved for the strength objective function is $R_{\text{index}} = 2.0 \cdot 10^7$, whilst the achieved values for the constraint functions over transverse stiffness and elastic symmetries are: $\mathrm{E}_y{}^A = 49468$ MPa and $I(\mathbf{P}(\mathbf{x})) = 7.74 \cdot 10^{-5}$.

The respect of elastic symmetries is confirmed by the graphics of polar variation of stiffness properties for laminate (30), shown in Fig. 11.

The effectiveness of the genetic algorithm BIANCA is proved by the graphics of average and best values of the objective function over the generations, shown in Fig. 12. Calculations were run over a population composed of $n = 400$ individuals and quasi-optimal individuals were obtained after less than 150 generations.

7.4 A multi-objective constrained optimisation problem: maximisation of first natural frequency and of in-plane principal stiffness with constraints over elastic symmetries

We show here an example of calculations run using BIANCA and dealing with a multi-objective optimisation problem for composite laminates: maximisation of the first natural frequency and maximisation of the principal in-plane Young modulus $E_x{}^A$. At the same time, as usual, we impose the necessary constraints over required elastic symmetries expressed by function $I(\mathbf{P}(\mathbf{x}))$. The corresponding formulation of the resulting multi-objective optimisation problem is given as:

$$\begin{aligned}
&\max\ (\min_{m,n} \omega_{mn}) \\
&\ \text{and} \\
&\max\ E_x^A \\
&\text{such that}: \quad I(\mathbf{P}(\mathbf{x})) < \varepsilon \quad (\varepsilon = 10^{-4})
\end{aligned} \tag{31}$$

where the expression of the natural frequencies for a rectangular simply supported plate made of an uncoupled orthotropic material is (ρ and h denote the mass density and the thickness of the plate, respectively):

$$\frac{\omega_{mn}}{\pi^2} = \sqrt{\frac{D_{11}(m/a)^4 + 2(D_{12} + 2D_{66})(m/a)^2(n/b)^2 + D_{22}(n/b)^4}{\rho h}} \tag{32}$$

The required elastic symmetries here, expressed by the function $I(\mathbf{P}(\mathbf{x}))$, are extension and bending orthotropy ($K_A = K_D = 0$) with coincident axes and uncoupling.

We ran the calculations using the combined strategy of *fitness sharing* and *niche* methods developed within BIANCA together with the *automatic dynamic penalisation* technique of constraint handling (see Sects. 4 and 5). We used a population composed of $n = 200$ individuals, and we obtained the Pareto front within 50 generations.

We fixed the number of layers $n = 10$, whilst the optimisation variables are the stacking sequence (which is completely free) and the corresponding values of the orientation angles. The angles can take all values between $-90°$ and $90°$, with a discrete precision of $p = 1°$.

In this case, the result of a run of BIANCA is the group of non-dominated individuals (Pareto front) belonging to the final generation, as shown in Fig. 13.

The designer can choose among the design solutions belonging to the Pareto front according to some additional criterion (feasibility, mechanical properties, ...). We give here an example corresponding to the point called "solution 1" in Fig. 13, which is situated at one end of the Pareto front: it shows the highest value of Young modulus $E_x{}^A$ and the smallest value of the fundamental frequency. Its stacking sequence is the following:

$$[14/-2/-25/-7/-1/-3/17/1/15/-15] \tag{33}$$

For laminate (33) the smallest fundamental frequency is $\omega_{11} = 33.21$ Hz and the principal in-plane Young modulus is $E_x^A = 159000$ MPa. Solution (33) satisfies as well the conditions over elastic symmetries, as it can be seen in Fig. 14.

7.5 BIANCA and other global optimisers for optimal design of composite laminates

To the largest knowledge of the authors, the literature in the field of optimal design of composite laminated structures is abundant in number of works published, and a lot of researchers developed genetic algorithms in order to solve the related problems (see for instance [6] and references therein). Nevertheless, the existing global optimisers are devoted to particular

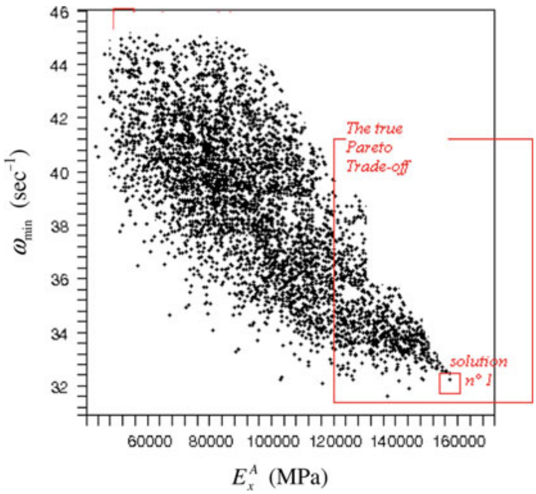


Fig. 13 Final population and Pareto front for problem (31)

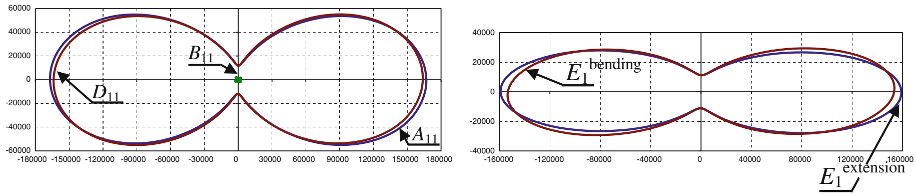


Fig. 14 Polar representation of the elastic properties for laminate (33)

problems of optimisation of composite laminates, without establishing a general approach. In addition, they seem to lack a rich genetic or evolutionary content:

1. as it is done in BIANCA, it is common use to represent discrete variables, because it is in the very nature of the problems to be solved. Yet, it is generally accepted to limit the set of admissible values to a very small number (for instance 0/±45/90 for the orientation angles) and, consequently, variables are encoded using a maximal alphabet instead of a minimal binary one, as we do in BIANCA.
2. It is also common use to apply the genetic operator of crossover (single- or multi-point) directly on the chromosome representing the whole stacking sequence. Thus the search is more like a sort of permutation strategy than a genetic algorithm. On the contrary, in BIANCA we apply the crossover on every single gene, thus assuring a deep recombination of the genetic information and a large exploration of the search space.
3. Finally, it is common practice to introduce a number of simplifying hypothesis in order to automatically respond to some design criteria, such as elastic uncoupling or orthotropy. Consequently, the complexity of the design of laminates is reduced by the limitation of the search space to a given class of candidate solutions, and that is at the expense of the generality of the design approach and of the number of potential solutions.

To the largest knowledge of the authors, there is no example in the literature of global optimisation of composite laminates using a commercially available code. Yet, an example of global optimiser applied to our formulation of the optimal design of laminated composite structures can be found in [20]. The approach is based on PSO (Particle Swarm Optimisation). The advantage is a higher speed of the algorithm, but the genetic approach in BIANCA still shows better performances in terms of the repeatability and the high quality of results (see [7,10,20]).

8 Conclusions

Often optimisation problems issued of engineering design are complex in terms of mathematical formulation, as they can be highly non linear and non convex [21–23].

In this paper, we presented the genetic algorithm BIANCA for solving optimisation problems in structural engineering. Inspired by the peculiar needs of the optimal design of composite laminated structures, we made an effort in order to create a very rich and adaptive numeric tool for more general problems in engineering.

The interesting features of BIANCA are the rich and exhaustive representation of individuals and the new numerical strategies devoted to constraint handling in the case of single- and multi-objective optimisation problems. All these points are described in this paper.

Finally, we give some examples of applications of BIANCA. Firstly, we presented some theoretical problems which we used to test the effectiveness of the new strategies developed within BIANCA. Secondly, we illustrated the case of the global optimisation of composite laminates, concerning both the theoretical formulation and the numerical results found by BIANCA. The large number of results found and their good precision show the robustness and the effectiveness of the genetic algorithm BIANCA when dealing with such complex combinatorial optimisation problems.

References

1. Haftka, R.T., Gürdal, Z.: Elements of structural optimization. Springer, Berlin (1992)
2. Arora, J.: Optimization of structural and mechanical systems. World Scientific, Singapore (2007)
3. Deb, K.: Multi-objective optimization using evolutionary algorithms. Wiley, London (2001)
4. Goldberg, D.: Genetic algorithms in search, optimization, and machine learning. Addison and Wesley, Reading (1989)
5. Michalewicz, Z.: Genetic algorithms + data structures = evolution programs. Spinger, Berlin (1996)
6. Gürdal, Z., Haftka, R.T., Hajela, P.: Design and optimization of laminated composite materials. Wiley, London (1999)
7. Vincenti, A.: Conception et optimisation de composites par méthode polaire et algorithmes génétiques. PhD Thesis, Université de Bourgogne, ISAT, 2002 (in French)
8. Vincenti, A., Vannucci, P., Verchery, G.: Design of composite laminates as an optimization problem: a new genetic algorithm approach based upon tensor invariants. In: Proceedings of WCSMO 5 (5th World Congress on Structural and Multidisciplinary Optimization). Lido di Jesolo-Venice, Italy, 19th–23th May 2003
9. Vannucci, P.: Designing the elastic properties of laminates as an optimisation problem: a unified approach based on polar tensor invariants. Struct. Multidiscip. Optim. **31**(5), 378–387 (2006)
10. Deb, K.: An efficient constraint handling method for genetic algorithms. Comput. Methods Appl. Mech. Eng. **186**, 311–338 (2000)
11. Coello Coello, C.A.: Theoretical and numerical constraint handling techniques used with evolutionary strategies: a survey of the state of the art. Comput. Methods. Appl. Mech. Eng. **191**, 1245–1287 (2002)
12. Landa Becerra, R., Coello Coello, C.A.: Cultured differential evolution for constrained optimization. Comput. Methods. Appl. Mech. Eng. **195**, 4303–4322 (2006)
13. Santana-Quintero, L. V., Hernàndez-Dìaz, A.G., Molina, J., Coello Coello, C.A.: DEMORS: a hybrid multi-objective algorithm using differential evolution and rough set theory for constrained problems. Comput. Oper. Res. (in press) (2009)
14. Habbal, A., Petersson, J., Thellner, M.: Multidisciplinary topology optimization solved as a Nash game. Int. J. Numer. Methods. Eng. **61**, 949–963 (2004)
15. Deb, K.: Introduction to evolutionary multiobjective optimization. Lect. Notes. Comput. Sci. **5252**, 59–96 (2008)
16. Marco, N., Desideri, J.A., Lanteri, S.: Multi-objective optimization in CFD by genetic algorithms. INRIA reports no. 3686 (1999)
17. Ahmadian, M.R.: A general strategy for the optimal design of laminated composites by the polar-genetic method. PhD thesis, Université de Versailles (2007)
18. Vincenti, A., Ahmadian, M.R., Vannucci, P.: A general strategy for the optimal design of laminated composites by the polar-genetic method. Int. J. Mech. Sci., (Submitted) (2009)
19. Park, W.J.: An optimal design of simple symmetric laminates under the first ply failure. J. Compos. Mater. **16**, 341–355 (1982)
20. Vannucci, P.: ALE-PSO: an adaptive swarm algorithm to solve design problems of laminates. Algorithms **2**, 710–734 (2009). doi:10.3390/a2020710
21. Pardalos, P.M., Resende, M.G.C.: Handbook of applied optimization. Oxford University Press, Oxford (2002)
22. Hirsch, M.J., Meneses, C.N., Pardalos, P.M., Resende, M.G.C.: Global optimization by continuous grasp. Optim. Lett. **1**(2), 201–212 (2007)
23. Hirsch, M.J., Pardalos, P.M., Resende, M.G.C.: Solving systems of nonlinear equations with continuous grasp. Nonlinear Anal. Real World Appl. **10**(4), 2000–2006 (2009)