



UNIVERSITÀ  
DEGLI STUDI  
FIRENZE



UNIVERSITÀ DI PISA



Università  
degli Studi  
di Perugia



Technische  
Universität  
Braunschweig

International Doctorate in Civil and Environmental Engineering

# Anisotropic Structures - Theory and Design

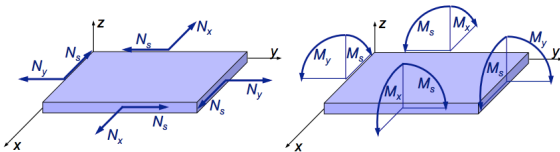
Strutture anisotrope: teoria e progetto

Paolo VANNUCCI

UNIVERSITÉ DE  
VERSAILLES  
ST-QUENTIN-EN-YVELINES



UNIVERSITÉ PARIS-SACLAY



Lesson 7 - June 4, 2019 - DICEA - Università di Firenze

## Topics of the seventh lesson

- The design of anisotropic laminated structures as an optimization problem



# About optimization techniques

Because the design of laminated anisotropic structures is a difficult task, the basic idea is to transpose it into an effective, proved theory: that of **mathematical optimization**.

In such a theoretical framework it is possible to give a **correct and clear mathematical formulation** of the design problems and also to find **effective numerical tools** for the resolution of such problems.

Before going on, it is worth to recall some basic aspects of optimization theory and introduce some numerical tools particularly suited for the problems of laminates design.

# Types of optimization problems

Generally speaking, an optimization problem can be always reduced to a form of the type

$$\begin{aligned} & \min_{x_i \in \Omega} f(x_i), \quad i = 1, \dots, n, \\ & \text{subjected to } g_j(x_i) \leq 0, \quad j = 1, \dots, p, \\ & \text{and to } h_k(x_i) = 0, \quad k = 1, \dots, q. \end{aligned} \tag{1}$$

with:

- $x_i$ : design variables
- $f(x_i)$ : objective (or cost) function
- $g_j(x_i)$ : inequality constraints
- $h_k(x_i)$ : equality constraints

The set  $\Omega$  of the points  $x_i$  that satisfy all the constraints is the **feasible domain**.

In several problems, a **state equation** is one of the equality constraints; it states a necessary condition to be satisfied by the solution, e.g. in mechanics the **equilibrium equation**, a condition that the optimal solution must satisfy (in some formulations, such a state equation can be put in variational form, for instance using the principle of minimum total potential energy, and it can enter directly the objective function, leading so to a double minimization problem).

When there are more than one conflicting objectives to be minimized, the problem is **multiobjective**.

A problem is **continuous** if  $x_i \in \mathbb{R} \forall i = 1, \dots, n$ , **discrete** if  $\exists i : x_i \notin \mathbb{R}$ .

# Convexity

Convexity is one of the most important characteristics of optimization problems.

The reason is that **convexity**  $\Rightarrow$  **uniqueness** of the minimum.

An optimization problem is convex  $\iff$  the objective function  $f(x_i)$  and the feasible domain  $\Omega$  are convex.

A domain  $\Omega$  is **convex**  $\iff \forall x_1, x_2 \in \Omega, x_1 \neq x_2,$

$$x = (1 - t)x_1 + t x_2 \in \Omega \quad \forall t \in [0, 1]. \quad (2)$$

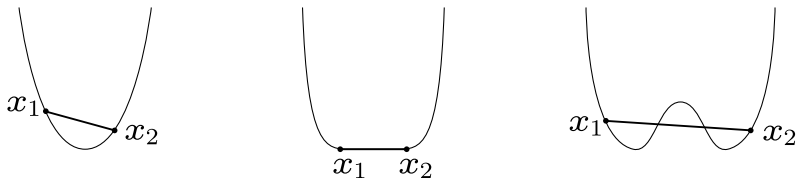
The intersection of two or more convex domains is a convex domain.

A function  $f(x_i) : \Omega \rightarrow \mathbb{R}$  is **convex on the convex domain**  $\Omega \subset \mathbb{R}^n$   
 $\iff \forall x_1, x_2 \in \Omega, x_1 \neq x_2$

$$f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2) \quad \forall \lambda \in (0, 1). \quad (3)$$

The function  $f(x_i)$  is **strictly convex** if strict inequality holds.

For  $n = 1$ , a convex function is below the line joining  $x_1$  and  $x_2$ .



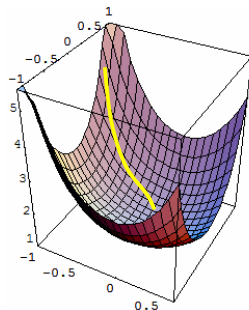
A strictly convex (*left*), a convex (*middle*), and a nonconvex (*right*) function

## Descent methods

The basic idea for convex problems is to start from a given point and to go down until the minimum.

Because the problem is convex, one arrives always to the minimum.

The descent methods make use of the derivatives of  $f(x_i) \Rightarrow$  it can be used only with functions at least  $C^1$ , so it cannot be used for discrete problems.



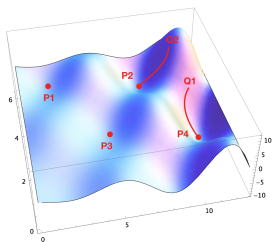
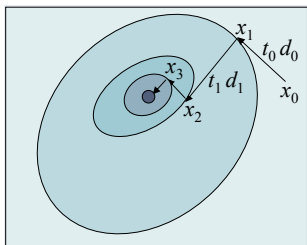
There are different descent methods; the basic one is the **steepest descent method**.

It is composed of different steps:

- choice of a feasible starting point  $x_i^0 \in \Omega$ ;
- for each step  $k$ , computation of the steepest descent direction  $d_i^k$ ;
- for each step  $k$ , computation of the step length  $t^k$ ;
- calculation of the new point  $x_i^{k+1} = x_i^k + t^k d_i^k$ .
- stop when  $d_i^k = 0 \forall i$ .

By the same properties of the gradient,  $d_i = -\nabla_i f$ .

The step length can be calculated by different methods: dichotomy, quadratic interpolation, golden section etc.

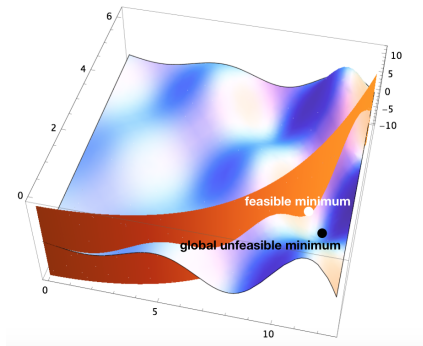


It can be shown that  $d^k \cdot d^{k+1} = 0 \rightarrow$  the method is not very effective.

Other descent methods improve the descent track: quasi-Newton, conjugate gradient etc.

Using descent methods with non-convex functions can lead to **local minima**: the starting point Q2 leads to the local minimum P2, while Q4 leads to the global minimum P4.





When constraints are imposed, the minimum can be on the boundary and it can correspond to points where the gradient is not null.

Different methods can be used to take into account for constraints: barrier, penalization etc.

# Metaheuristics

The true drawback of descent methods is the **sensitivity to the initial point**

If the problem is non-convex, the convergence to a true global minimum is not guaranteed.

To counter this problem, the basic idea is to work not with a unique point, but with a **population of individuals**.

An individual is a vector  $x_i$ ,  $i = 1, \dots, n$  candidate to be a solution of the problem.

Letting evolve a population of potential solutions help in reaching the global minimum.

The dynamics that inspires the displacement of the population throughout the feasible domain is called a **metaheuristic**.

A metaheuristic is hence a rationale inspired by a given phenomenon that can be **biological, social or physical**.

There are, in fact, several different metaheuristics:

- simulated annealing, inspired by metallurgy of alloys;
- ant colonies, inspired by the social dynamics of ants;
- neural networks, inspired by the brain functioning;
- tabu search, inspired by social rules;
- genetic algorithms, inspired by Darwinian selection;
- particle swarm optimization (PSO), inspired by the dynamics of flocks of birds or shoals of fishes.

Metaheuristics are **order zero methods**: they do not need the calculation of the derivatives of the objective function.

As such, they can be used also with discrete problems.

# Genetic algorithms

Genetic Algorithms (GAs) are perhaps the most used metaheuristic.

GAs were introduced by Holland in 1965 and are inspired to the mechanism of **natural selection** (C. Darwin, The origin of species, 1859).

The basic idea is that of letting evolve a population following the rules of the survival of the fittest (Darwinian selection).

The dynamics of the changes of the population from a generation, i.e; an iteration of the algorithm, to the following one are based upon the laws of genetics.

The success of GAs is mainly due to their **robustness and effectiveness** in dealing with non convex problems.

The general scheme of a classical GA is rather simple:

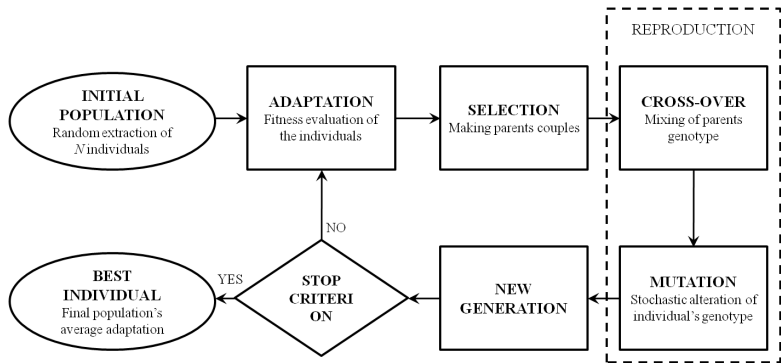


Figure: General scheme of a classical GA

The **adaptation** of each individual is an operation aiming at giving a ranking of individuals with respect to the objective: better individuals, i.e. individuals for which the objective has a better value, have a better **fitness**  $\varphi$ .

There are different ways to introduce a fitness, normally  $\varphi \in [0, 1]$ : 0 corresponds to the worst individual, 1 to the fittest one.

A classical way is to define the fitness as

$$\varphi = \left( 1 + \frac{f - \min_{pop} f}{\min_{pop} f - \max_{pop} f} \right)^c, \quad c \geq 1. \quad (4)$$

Coefficient  $c$  is used to tune the selection pressure.

Defined in this way,  $0 \leq \varphi \leq 1$ .

The Darwinian idea is that individuals with a better fitness, i.e. best adapted to environment, have a greater chance to **survive a longer time**, so to reproduce themselves.

Hence, in the **selection** phase best fitted individuals have a greater chance to be selected.

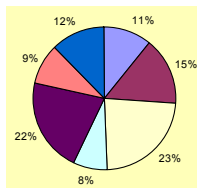
There are different ways to operate selection; in all the cases, a couple of individuals, **the parents**, is selected to be reproduced in the subsequent phases.

Two widely used selection strategies are **tournament** and **roulette wheel**.

In the tournament selection,  $k$  individuals are randomly chosen and only the best one is selected; doing this  $N$  times, one gets  $N$  parents; these are then coupled randomly to generate  $N$  offsprings.

In roulette wheel selection,  $N/2$  couples are randomly selected on the base of their fitness: the higher the fitness  $\varphi_i$  of individual  $i$ , the higher its probability  $p_i$  to be selected for reproduction:

$$p_i = \frac{\varphi_i}{\sum_{j=1}^N \varphi_j}. \quad (5)$$





## Coding individuals: phenotypes and genotypes

In a classical GA, each real or discrete variable, the **phenotype**, is coded binary to obtain its **genotype**.

It is hence represented by a chain of 0s and 1s: the DNA chain of an individual.

The genetic operations are done on the binary chains of two individuals, the **parents**, to obtain (hopefully) two better **offsprings**.

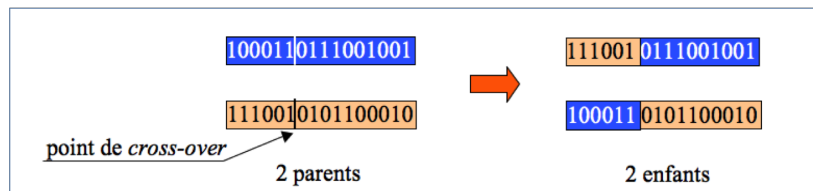
There are two classical genetic operations: **crossover** and **mutation**.

both of them are used to generate offsprings from selected parents; because of selection, there is a genetic improvement of the population, i.e., the objective function is **globally improved**.

In this way, from a generation to the following one, more and more fitted individuals form the populations → the probability to have better individuals, i.e. phenotypes approaching the minimum, increases.

# Genetic operations

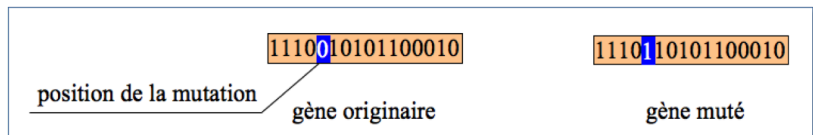
Classical binary operations are done on the genotypes, i.e. on the binary strings. Cross-over on genes (1 gene=1 design variable)



Cross-over allows to obtain 2 offsprings from two selected parents.

The crossover point is randomly selected

## Gene mutation:



Mutation is introduced to guarantee a certain biodiversity, which mathematically speaking serves to better explore the feasible domain and avoid local minima (premature convergence).

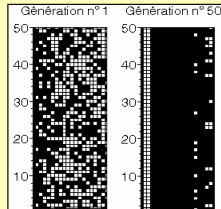
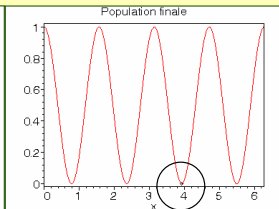
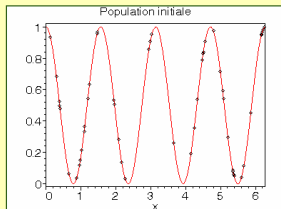
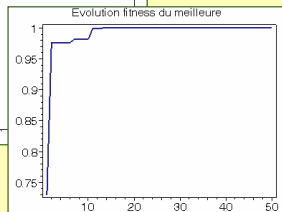
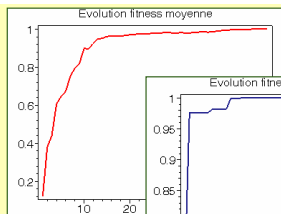
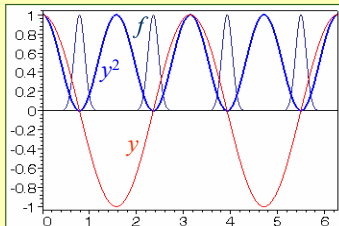
The binary digit to be mutated is randomly selected.

A third operator is often used: [elitism](#). It consists in preserving the best individual of each generation, that replaces the worst one of the offsprings.

## An exemple of non convex function

Problem: to solve the equation  $y = \cos 2x$  in  $[0, 2\pi]$ . The problem can be seen as: find the minimum of  $y^2$ .

$$y = \cos 2x \text{ dans } [0, 2\pi]$$



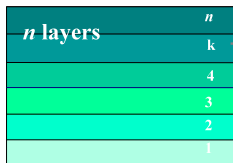
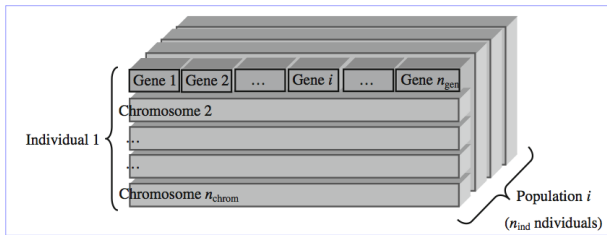
# The GA BIANCA

BIANCA (Biologically Inspired ANalysis of Composite Assemblages) is a GA with some special characteristics, specially conceived for highly non convex constrained modular problems

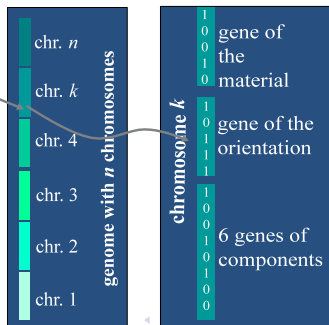
Its main features are

- multi-chromosome and multi-gene
- multi-population with migration operator
- virtual binary coding
- Boolean genetic operations over each gene
- elitism
- constraints handled by ADP (Automatic Dynamic Penalization) method
- it makes evolve simultaneously **species and individuals**, thanks to special crossing and mutation operators
- it can be interfaced, in principle, with any other code (namely ABAQUS, ANSYS, MATLAB etc.)

BIANCA has a special structure of the information that allows for a deep mixing of genomes and for the **simultaneous selection of species and individuals**:



**Remarque:** stratifiés avec  $n$  différent appartiennent à différentes espèces



Special genetic operators are introduced in BIANCA for the selection of the species.

Different species are characterized by a different number of chromosomes.

This is done to deal with new problems like those concerning modular systems.

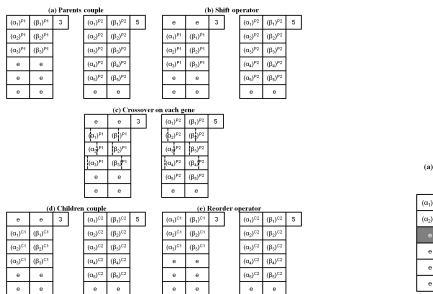


Figure 1.8: Crossover among species: (a) parents couple, (b) effect of the *shift* operator, (c) crossover on homologous genes, (d) children couple and (e) effect of the *chromosome reorder* operator.

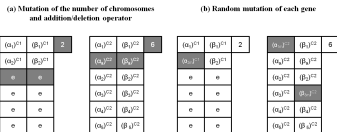


Figure 1.9: Mutation of species: (a) mutation of the number of chromosomes and effect of the *chromosome addition-deletion* operator, (b) effect of the mutation operator on every gene

## An exemple of non convex constrained function

Problem: find the minimum of

$$f(x_1, x_2) = -e^{ka\sqrt{x_1^2+x_2^2}} \sin ax_1 \cos 2bx_2, \quad (6)$$

in the domain

$$\Omega = \{(x_1, x_2) : 0 \leq x_1 \leq 4\pi, 0 \leq x_2 \leq 2\pi\}, \quad (7)$$

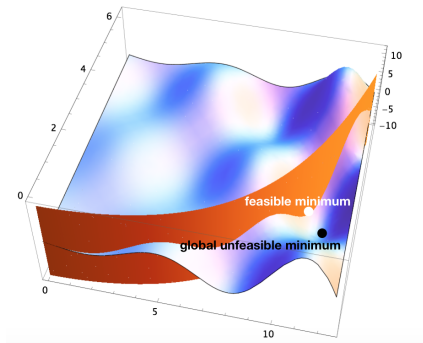
with the constraint

$$g(x_1, x_2) = e^{cx_1^2} - x_2 - 1 \leq 0, \quad (8)$$

with  $a = 1.$ ,  $b = 0.6$ ,  $c = 0.012$ ,  $k = 0.2$ .



The function is non convex, different local minima exist in  $\Omega$  but the absolute admissible minimum is on the boundary.

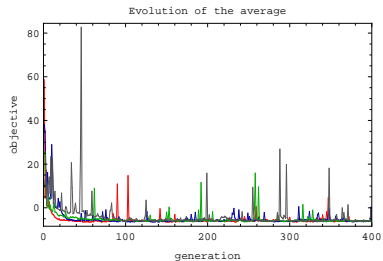
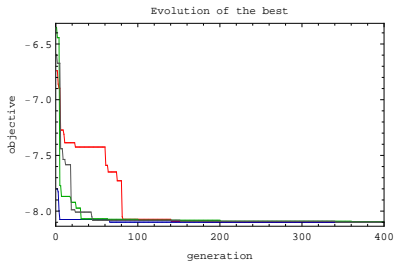


The GA BIANCA has been used with the following parameters:

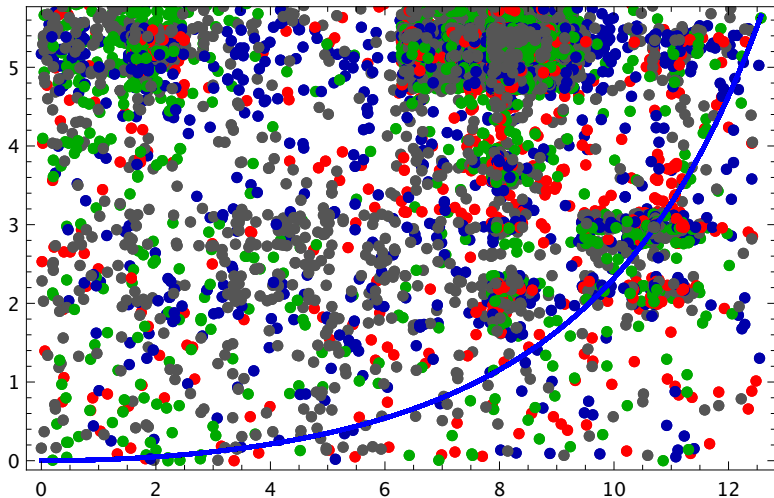
- 4 populations, each one of 200 individuals
- 400 generations
- probability of cross-over: 0.8
- probability of mutation: 0.04
- isolation time: 20 generations
- elitism
- selection type: roulette wheel

Best solution:  $p = (10.711, 2.966)$ ,  $f = -8.099$ .

## Convergence diagrams



## Dynamics of the populations



# The PSO code ALE-PSO

The code ALE-PSO is a Particle Swarm Optimization code is a classical PSO code (Eberhart and Kennedy, 1995), but with adapting coefficients, that evolve through a prescribed scheme, specified by the operator

PSO are inspired by and they mimic in an algebraic way. the social behavior and dynamics of groups of individuals (particles), such as the flocks of birds, whose groups displacements are not imposed by a leader: the **same overall behavior of the flock guides itself**.

Their main advantage is their true simplicity: the updating rule is

$$\begin{aligned} \mathbf{u}_{t+1}^k &= r_0 c_0 \mathbf{u}_t^k + r_1 c_1 (\mathbf{p}_t^k - \mathbf{x}_t^k) + r_2 c_2 (\mathbf{p}_t^g - \mathbf{x}_t^k) \\ \mathbf{x}_{t+1}^k &= \mathbf{x}_t^k + \mathbf{u}_{t+1}^k, \quad k = 1, \dots, m, \quad t = 1, \dots, s \end{aligned} \quad (9)$$

- $\mathbf{x}_{t+1}^k$ : vector representing the position, in the  $n$ -dimensional problem space, of the  $k$ -th particle in a swarm of  $m$  particles, at the time-step  $t + 1$  of  $s$  total steps;
- $\mathbf{u}_{t+1}^k$ : displacement (often called velocity) of the particle  $\mathbf{x}^k$  from its position  $\mathbf{x}_t^k$  at the time step  $t$  to its updated position  $\mathbf{x}_{t+1}^k$  at the time-step  $t + 1$ ;
- $\mathbf{p}_t^k$ : vector recording the best position occupied so far by the  $k$ -th particle (personal best position); in a minimization problem for the objective function  $f(\mathbf{x})$ ,  $\mathbf{p}_t^k$  is updated as follows:

$$\mathbf{p}_{t+1}^k = \begin{cases} \mathbf{p}_t^k & \text{if } f(\mathbf{x}_{t+1}^k) \geq f(\mathbf{p}_t^k) \\ \mathbf{x}_{t+1}^k & \text{if } f(\mathbf{x}_{t+1}^k) < f(\mathbf{p}_t^k) \end{cases} \quad (10)$$

- $\mathbf{p}_t^g$  : vector recording the best position occupied so far by any particle in the swarm (global best position); in a minimization problem for the objective function  $f(\mathbf{x})$ ,  $\mathbf{p}_t^g$  is updated as follows:

$$\mathbf{p}_{t+1}^g \in \{\mathbf{p}_{t+1}^k, k = 1, \dots, m\} \text{ such that} \quad (11)$$

$$f(\mathbf{p}_{t+1}^g) = \min_{k=1, \dots, m} f(\mathbf{p}_{t+1}^k)$$

- $r_0, r_1, r_2$ : independent random coefficient uniformly distributed in  $[0, 1]$
- $c_0, c_1, c_2$ : inertial, cognition and social parameters

In ALE-PSO  $c_0, c_1, c_2$  are updated during iterations using a power law

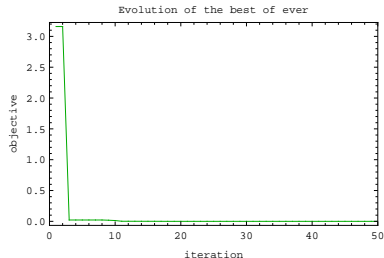
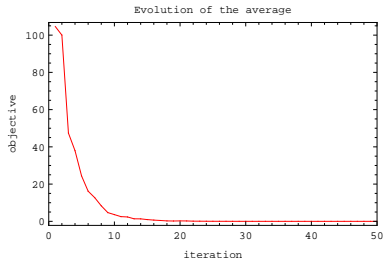
## An exemple of unconstrained function

Problem: find the minimum of

$$y = x_1^2 + x_2^2 + x_3^2. \quad (12)$$

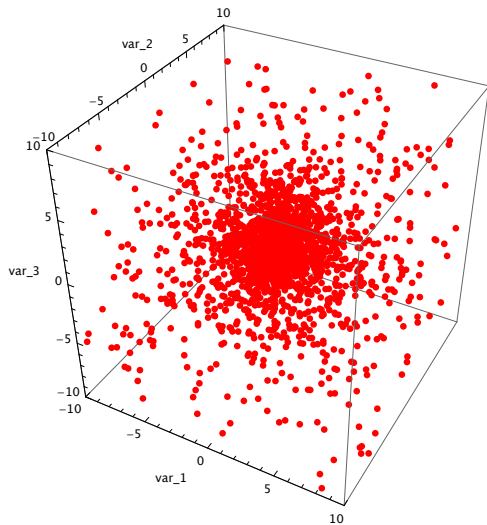
We have used AE-PSO with a swarm of 200 particles and with 50 iterations.

The best solution is found after 25 iterations.





## The dynamics of the swarm:



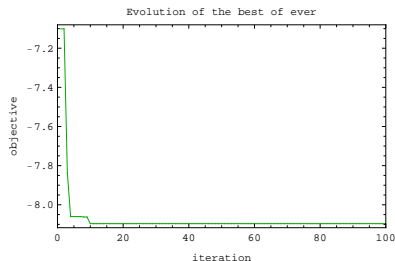
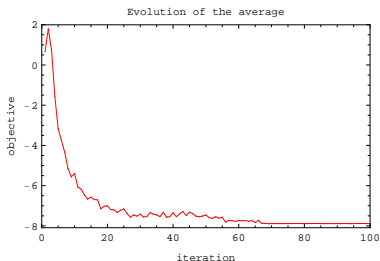
## An exemple of constrained function

We solve with ALE-PSO the same problem previously solved with BIANCA, eq. (6).

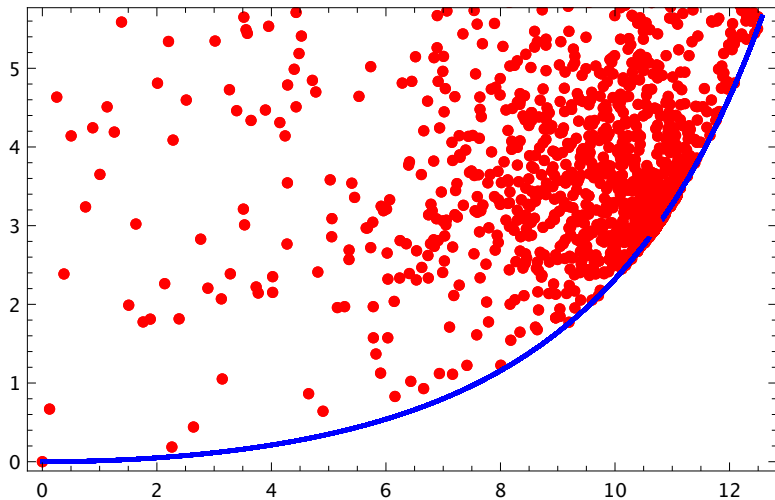
The constraint is handled through a death-penalty method.

We have used a swarm of 100 particles for 100 iterations.

The best solution is  $p = (10.699, 2.954)$ , with  $f = -8.096$  found after 10 iterations.



The dynamics of the swarm:



# Optimisation of anisotropic laminates

- Typical problems:

$$\min_x f(x) \quad (13)$$

$x$ : design variables (typically:  $n$ ,  $\delta_j$ , thicknesses etc.)  
usually, the material is chosen *a priori*  $\rightarrow$  the isotropic part is completely determined by this choice

# Optimisation of anisotropic laminates

- Typical problems:

$$\min_x f(x) \quad (13)$$

$x$ : design variables (typically:  $n$ ,  $\delta_j$ , thicknesses etc.)  
usually, the material is chosen *a priori*  $\rightarrow$  the isotropic part is completely determined by this choice

- Be:

$$\mathcal{P} = \{\mathcal{P}_i, i = 1, \dots, 12\} = \{R_0, R_1, \Phi_0 - \Phi_1, \Phi_1\}_{A,B,D} \quad (14)$$

$T_0$  and  $T_1$  do not appear because we consider here only laminates with identical plies

# Optimisation of anisotropic laminates

- Typical problems:

$$\min_x f(x) \quad (13)$$

$x$ : design variables (typically:  $n$ ,  $\delta_j$ , thicknesses etc.)  
usually, the material is chosen *a priori* → the isotropic part is completely determined by this choice

- Be:

$$\mathcal{P} = \{\mathcal{P}_i, i = 1, \dots, 12\} = \{R_0, R_1, \Phi_0 - \Phi_1, \Phi_1\}_{A,B,D} \quad (14)$$

$T_0$  and  $T_1$  do not appear because we consider here only laminates with identical plies

- Key points:

$\mathbb{A} = \mathbb{A}(\mathcal{P}_i)$ ,  $\mathbb{B} = \mathbb{B}(\mathcal{P}_i)$ ,  $\mathbb{D} = \mathbb{D}(\mathcal{P}_i)$ ,  $i = 1, \dots, 12$ ,  
unique correspondence

# Optimisation of anisotropic laminates

- Typical problems:

$$\min_x f(x) \quad (13)$$

$x$ : design variables (typically:  $n$ ,  $\delta_j$ , thicknesses etc.)  
usually, the material is chosen *a priori*  $\rightarrow$  the isotropic part is completely determined by this choice

- Be:

$$\mathcal{P} = \{\mathcal{P}_i, i = 1, \dots, 12\} = \{R_0, R_1, \Phi_0 - \Phi_1, \Phi_1\}_{A,B,D} \quad (14)$$

$T_0$  and  $T_1$  do not appear because we consider here only laminates with identical plies

- Key points:

$\mathbb{A} = \mathbb{A}(\mathcal{P}_i)$ ,  $\mathbb{B} = \mathbb{B}(\mathcal{P}_i)$ ,  $\mathbb{D} = \mathbb{D}(\mathcal{P}_i)$ ,  $i = 1, \dots, 12$ ,  
unique correspondence  
functions

$$\mathcal{P}_i = \mathcal{P}_i(\delta_j), \quad i = 1, \dots, 12, \quad j = 1, \dots, n \quad (15)$$

are not bijective

# An optimisation *naturally* sequential

- **Step 1:** the structure problem

$$\text{look for } \mathcal{P}_i = \mathcal{P}_i^{\text{opt}} \in \mathcal{R} \subset \mathbb{R}^{12} : \quad (16)$$

$$f(\mathcal{P}_i^{\text{opt}}) = \min_{\mathcal{P}_i \in \mathcal{R}} f(\mathcal{P}_i) \quad (17)$$



# An optimisation *naturally* sequential

- **Step 1:** the structure problem

$$\text{look for } \mathcal{P}_i = \mathcal{P}_i^{\text{opt}} \in \mathcal{R} \subset \mathbb{R}^{12} : \quad (16)$$

$$f(\mathcal{P}_i^{\text{opt}}) = \min_{\mathcal{P}_i \in \mathcal{R}} f(\mathcal{P}_i) \quad (17)$$

- **Step 2:** the constitutive law problem

$$\text{find } \delta_j^{\text{sol}} \in \Delta \subset \mathbb{R}^n : \quad (18)$$

$$\varphi(\delta_j^{\text{sol}}) = \min_{\delta_j \in \Delta} \varphi(\delta_j), \quad \text{with} \quad (19)$$

$$\varphi(\delta_j) = \sum_{k=1}^{12} (\mathcal{P}_i(\delta_j) - \mathcal{P}_i^{\text{opt}})^2 \quad (20)$$

Some remarks on the structure problem:

- it has a mathematical structure that is **problem dependent**: the type of the objective functional depends upon the problem
- some **constraints** are normally part of it, specifying some requirements of different type: technical, mathematical, mechanical etc.
- the **geometrical bounds** are constraints that are always to be added to it; this is needed to have optimal properties that can be really obtained by a laminate
- because the geometrical bounds are convex but non linear, it is **never linear but it can be convex**
- its dimension is problem dependent; anyway, in the case of design of orthotropic and constant  $\mathbb{A}$  or  $\mathbb{D}$ , its dimension is reduced to only **3 design variables** if the polar formalism is used

Some remarks on the constitutive law problem:

- it is always **highly non convex**, due to the dependence of the elastic tensors upon the orientations (fourth powers of circular functions of the orientations)
- its dimension is equal to  $n - 1$
- it can be a **continuous or a discrete** one
- some constraints can be added to it

While the numerical approach for solving the structure problem is not unique and special procedures can exist for different problems, the best way for solving the constitutive law problem is always a **metaheuristic**: genetic algorithms (GAs), particle swarm optimization (PSO), ant colonies, (AC), simulated annealing (SA) and so on

# The constitutive law problem

The objective is to have a general approach for designing laminates with some given properties in a **completely general way**, i.e. without simplifying assumptions

In fact, only in this way it is possible to obtain a **true optimal solution**

The idea is to formulate the problem as a **problem of minimal distance of tensors**: the optimal laminate is the one whose tensors have a distance equal to zero from the target tensors

The target tensors have been determined previously, in some way

Typically, for simple problems, they can be found easily, while generally speaking they are determined as a solution of the structure problem

Among the properties to be designed, there are the elastic symmetries, typically orthotropy etc, uncoupling, quasi-homogeneity, thermal properties etc.

As we have already seen, when only the elastic properties are to be optimized, then the problem can be put in the form

$$\text{find } \delta_j^{sol} \in \Delta \subset \mathbb{R}^n : \quad (21)$$

$$\varphi(\delta_j^{sol}) = \min_{\delta_j \in \Delta} \varphi(\delta_j), \quad \text{with} \quad (22)$$

$$\varphi(\delta_j) = \sum_{k=1}^{12} (\mathcal{P}_i(\delta_j) - \mathcal{P}_i^{opt})^2 \quad (23)$$

The set  $\Delta$  of admissible orientations  $\delta_j$  depends upon the problem at hand and it can be continuous, regularly or irregularly discrete

The design variables are the orientations  $\delta_j, j = 2, \dots, n; \delta_1 = 0$  to fix the frame

The problem is highly **non convex** and the solution **never unique**

This is important, because it allows to put some constraints on the second and not on the first step

## Examples of design of the constitutive law

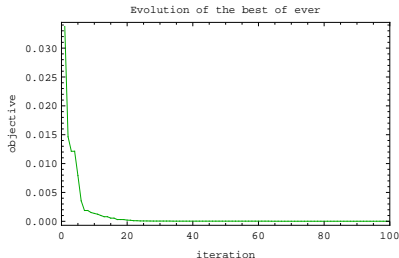
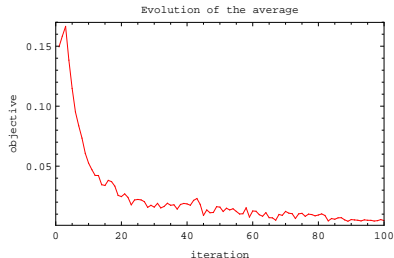
Example 1: 10-ply T300/5208 carbon-epoxy laminate with:

- $\mathbb{B} = 0 \Rightarrow R_0^B = R_1^B = 0$
- $R_0^A = R_0^D = 0$
- discrete angles (each  $1^\circ$ ).

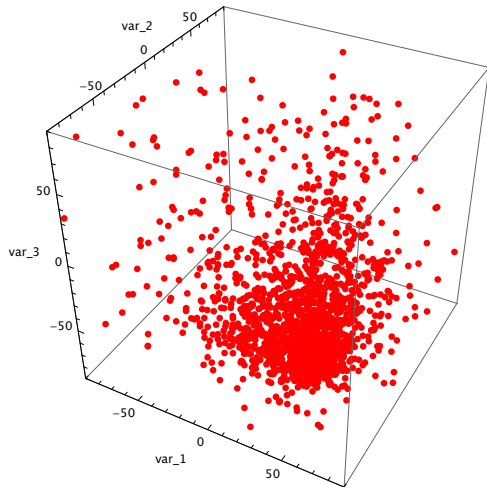
The best solution found by ALE-PSO (swarm of 100 particles for 100 iterations) is

$[31^\circ / -22^\circ / 90^\circ / -24^\circ / 19^\circ / -55^\circ / 90^\circ / -37^\circ / 44^\circ / 0^\circ]$

Residual:  $0.11 \times 10^{-4}$ .

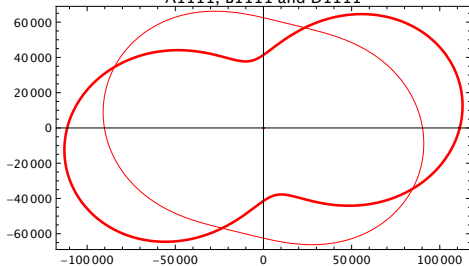


## The dynamics of the swarm:

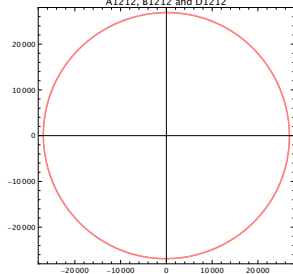


# The directional diagrams of the Cartesian components

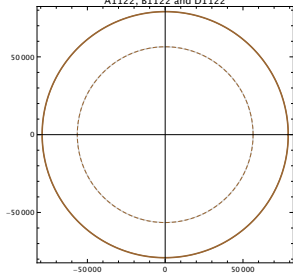
A1111, B1111 and D1111



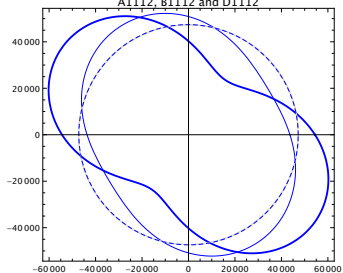
A1212, B1212 and D1212



A1122, B1122 and D1122



A1112, B1112 and D1112





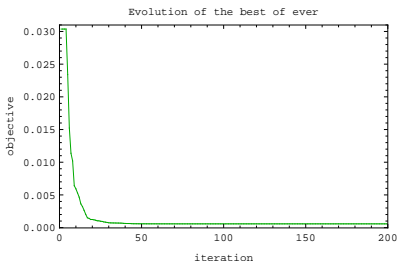
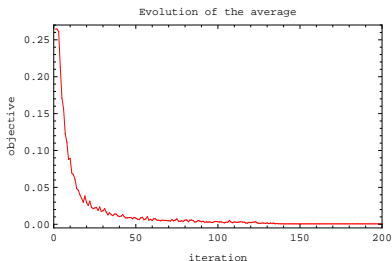
**Example 2:** a 12-ply carbon-epoxy fully isotropic uncoupled laminate.

- $\mathbb{B} = 0 \Rightarrow R_0^B = R_1^B = 0$
- $R_0^A = R_0^D = R_1^A = R_1^D = 0$

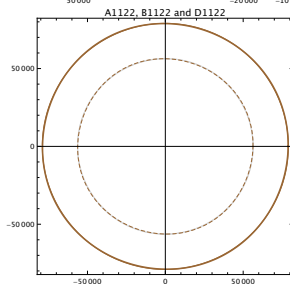
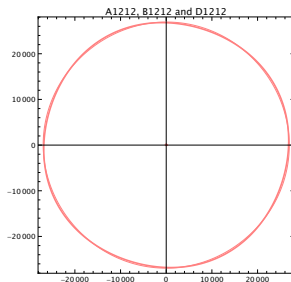
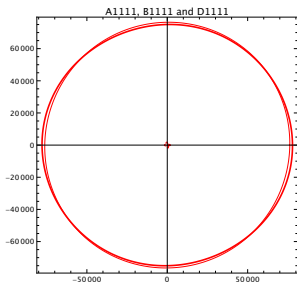
The best solution found by ALE-PSO (swarm of 200 particles for 200 iterations) is

$[-67.3^\circ/54.3^\circ/0.8^\circ/ -18.7^\circ/43.7^\circ/90^\circ/ -24.3^\circ/$   
 $-69.6^\circ/32.2^\circ/ -69.1^\circ/55.5^\circ/ -10.4^\circ]$

Residual:  $0.57 \times 10^{-3}$ .



# The directional diagrams of the Cartesian components

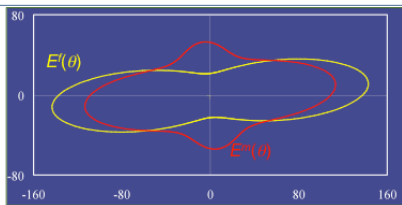
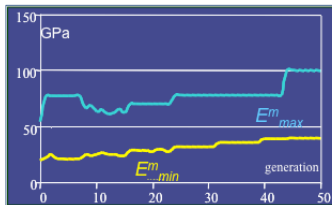


Example 3: 12-ply T300/5208 carbon-epoxy laminate with:

- $\mathbb{A}$  ordinarily orthotropic  $\Rightarrow \Phi_0^A - \Phi_1^A = K^A \frac{\pi}{4}$
- $\mathbb{B} = 0 \Rightarrow R_0^B = R_1^B = 0$
- $\delta_j \in \{0^\circ, 15^\circ, 30^\circ, 45^\circ, 60^\circ, 75^\circ, 90^\circ, \dots\}$
- $E_{max}^m \geq 100 \text{ GPa} (= 0.55 E_1)$
- $E_{min}^m \geq 40 \text{ GPa} (= 3.88 E_2)$

Best solution found by GA BIANCA:

$[0^\circ/30^\circ/15^\circ/15^\circ/90^\circ/75^\circ/0^\circ/45^\circ/75^\circ/0^\circ/15^\circ/15^\circ]$

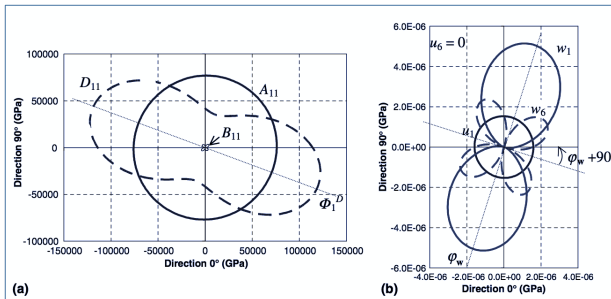


**Example 4:** 12-ply T300/5208 carbon-epoxy laminate with:

- $\mathbb{A}$  isotropic  $\Rightarrow R_0^A = R_1^A = 0$
- $\mathbb{D}$  ordinarily orthotropic with  $K^D = 0 \Rightarrow \Phi_0^D = \Phi_1^D$
- $\mathbb{B} = 0 \Rightarrow R_0^B = R_1^B = 0$
- $\delta_j \in \{-90^\circ, 90^\circ\}$
- isotropic tensor of thermal expansion coefficients
- cylindrical bending under a thermal gradient

Best solution found by GA BIANCA:

$[0^\circ / -29.9^\circ / 44.3^\circ / -61.8^\circ / 89.3^\circ / 61.8^\circ / 31.5^\circ / -89.1^\circ / 33.4^\circ / -71.7^\circ / -11.6^\circ / -28.1^\circ]$

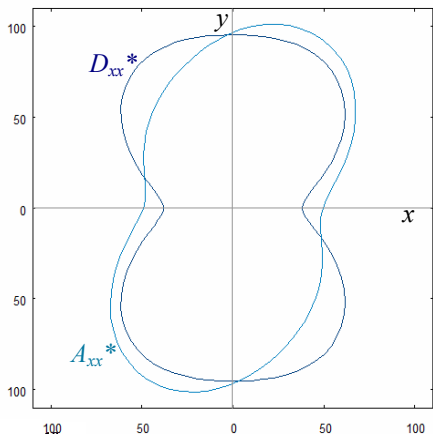


**Example 5:** 12-ply T300/5208 carbon-epoxy rectangular simply supported laminate with:

- $\mathbb{B} = 0 \Rightarrow R_0^B = R_1^B = 0$
- $\mathbb{D}$  ordinarily orthotropic  $\Rightarrow \Phi_0^D - \Phi_1^D = K^D \frac{\pi}{4}$
- Orthotropy axes aligned with the plate's sides  $\Rightarrow \Phi_1^D = 0$
- Frequency of the first mode  $\omega_1 \geq 150$  Hz

The best solution found by the code ALE-PSO is

$[-66.1^\circ / 66.5^\circ / 44.8^\circ / -36.4^\circ /$   
 $48.5^\circ / -5.7^\circ / 77.6^\circ / -90^\circ /$   
 $-84.5^\circ / 38.5^\circ / 57.5^\circ / -46^\circ /]$



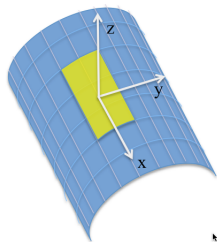
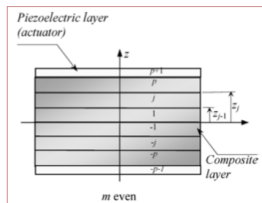
# A complete problem: structure plus laminate problem

This is an example of a complete case: the structure problem followed by the research of a suitable laminate

**Problem:** maximize the bending of a laminate produced by PZT actuators.

## Requirements:

- cylindrical bending
- $\mathbb{B} = 0 \Rightarrow R_0^B = R_1^B = 0$
- $\mathbb{D}$  ordinarily orthotropic  
 $\Rightarrow$   
 $\Phi_0^D - \Phi_1^D = K^D \frac{\pi}{4}$
- $\delta_j$  discretized at  $5^\circ$



Application to a T300/5208 carbon-epoxy 16-ply laminate

The structure problem: it can be formulated as follows ( $\mathcal{D} = \mathbb{D}^{-1}$ ):

$$\begin{aligned} \max_{\delta_j} f(\delta_j) &= \mathcal{D}_{12} + \mathcal{D}_{22} \\ \text{with} & \\ g(\delta_j) &= \mathcal{D}_{12} + \mathcal{D}_{11} = 0 \end{aligned} \tag{24}$$

The solution of the structure problem in this case can be found by hand and gives to the constitutive law problem the optimal values  $R_0^D$  and  $R_1^D$

The constitutive law:

- $R_0^D = R_0^D$
- $R_1^D = R_1^D$
- $R_0^B = 0$
- $R_1^B = 0$
- $\Phi_0^D - \Phi_1^D = K^D \frac{\pi}{4}$

$$\begin{aligned} \min_{\delta_j} \varphi(\delta_j) &= (R_0^D - R_0^D)^2 + (R_1^D - R_1^D)^2 + \\ & \rightarrow R_0^B + R_1^B + (\Phi_0^D - \Phi_1^D - K^D \frac{\pi}{4})^2 \end{aligned} \tag{25}$$

Best solution found by code ALE-PSO:

$[-20^\circ/5^\circ/0^\circ/15^\circ/15^\circ/5^\circ/40^\circ/45^\circ/90^\circ/$   
 $-10^\circ/20^\circ/ -15^\circ/10^\circ/5^\circ/20^\circ/ -15^\circ]$

