A Bilevel Integer Programming Method for Blended Composite Structures

S. Zein[1], M. Bruynëel[2]

Abstract

This paper proposes a new approach for the design of a composite structure. This approach is formulated as an optimization problem where the weight of the structure is minimized such that a reserve factor is higher than a threshold. The thickness of each region of the structure is optimized together with its stacking sequence and the ply drop-offs. The novelty of this approach is that, unlike in common practice, the optimization problem is not simplified and split into two steps, one for finding the thicknesses and one for the stacking sequence. The optimization problem is solved without any simplification assumption. It is formulated as a bilevel integer programming and it uses the backtracking procedure to satisfy the blending and the manufacturing rules. Some numerical experiments are performed to show the efficiency of the proposed optimization method over complex cases which cannot be solved with the existing methods.

Keywords: Integer Programming, Bilevel Optimization, Backtracking, Composite Structures.

1. Introduction

In the recent years, composite materials have taken a growing importance in the aeronautical industry. Because they exhibit high performance properties and lead to a considerable weight reduction, they can be an alternative choice in the design of many aircraft parts. The design and manufacturing processes of a panel are based on a ply drop-off technique. If the panel is divided into regions (Figure 1), each ply does not necessarily cover all the regions but some regions of it. As a consequence, the panel has a varying thickness over the surface of the panel which leads to a weight reduction. The fiber orientation in each ply can take one of the following values: $\{-45^o, 0^o, 45^o, 90^o\}$. The angle sequences of each region of the panel have to satisfy the blending and the manufacturing rules. The angle sequences must contain a fixed number of plies of each orientation, two consecutive angles cannot have a difference of 90^o, there can be at most four consecutive identical orientations and the sequences must be symmetric. These constraints are called the design rules. The continuity of the plies between two adjacent regions of the panel is referred to as the manufacturing rules. The nature of such rules make the optimization problem a combinatorial one.

[1] LMS Samtech HQ, Angleur (Liege), Belgium, samih.zein@lmsintl.com

[2] LMS Samtech HQ, Angleur (Liege), Belgium, michael.bruyneel@lmsintl.com

Many optimization methods based on genetic algorithms have been developed to address this specific problem. They differ by the technique which is used to satisfy the design/manufacturing rules. In [1, 2, 3, 4, 5, 6], the manufacturing constraints are addressed using a penalty approach. In [7, 8] a sub-laminate approach is used where regions sharing the same sub-laminates are grouped into one design variable. This method can guarantee the continuity of the plies in all the regions (blended structure).

In [9, 10, 11, 12], the continuity of the plies (the blending) is satisfied by a guide-based design. It gives blended structures but it does not provide a lot of flexibility in the design of the panel: the stacking sequence of the thickest region imposes the stacking sequences of the all other regions. For a given stacking guide, only one sequence can be assigned to each region. To overcome this difficulty, a general definition of the blended sequences design is proposed in [13]. The blending constraint is taken into account with a penalty approach. However, severe stress concentration can be observed when the panel is not blended. The authors of this paper found that the penalty approach is not an efficient choice for satisfying the blending constraints. Another blending approach is the one described in [14, 15], where the sequences of the regions are arranged into sets of plies which satisfy the blending principles. The approach in these two papers have the advantage of using the lamination parameters to compute the buckling instead of running expensive finite elements analysis.

In [16], the stacking sequence optimization problem is formulated as a linear integer programming problem where the orientation of each ply is modeled with four binary variables. The purpose of using these binary variables is to derive a mathematical expression of the manufacturing constraints. A linear expression of the buckling load is derived with respect to the stacking sequence in the case of a panel with one region. The paper studied the case of a panel of one region with eight unknown plies to be optimized. This approach is limited to the case of such a panel. It cannot handle the general case of a panel with regions of different thicknesses, like in Figure 1, because no linear expression for the buckling load can be derived. The same drawbacks have been found with the topology optimization approach proposed in [17, 18, 19]. They are able to optimize the buckling load with the manufacturing constraints but for a fixed blending scheme.

In [20], the authors have proposed a combinatorial method to optimize a buckling load with respect to the stacking sequence guide and the ply drop-offs, but the thicknesses were constant. This method has also been compared with other existing methods in [21] and in the case of single stacking sequences without blending. In this paper, the design of a composite panel is formulated as a bilevel integer programing where the weight of the panel is minimized subject to the buckling load higher than a safety threshold. The thicknesses of the regions are expressed in number of plies and they are updated together with the stacking sequence guide and the ply drop-off. The manufacturing and design rules are satisfied using the backtracking approach proposed in [20]. This approach does not use any approximation of it and can be generalized to a reserve factor of any type. The advantage of this approach over the classical one is discussed in the next section. The proposed algorithm in the paper only deals with the laminates of the structure and it cannot deal with other kind of composites. Therefore, the other parts of the structure remain constant during the optimization of the laminates. This is a limitation of the algorithm. If one is interested in optimizing the other parts, this must be after the optimization of the laminates and using other methods.

2. The Two-Step Optimization

The full design of a composite panel consists in minimizing the weight of the panel and satisfying some buckling load constraints. This task is commonly divided into two steps, like in [14].

- First step: this step gives a global description of the panel without the details on the stacking sequences. The computation of the buckling loads is based on the following assumption. Four different orientations are considered and the stacking sequence of each region is divided into four parts, each part is associated to an orientation and has its own thickness (see Figure 2). The design variables are the thicknesses per orientation and per region. They are continuous variables. Gradient based optimization methods over a finite elements code can be used to solve this problem.

- Second step: From the resulting thicknesses per orientation and per region of the preceding step, the number of plies of each orientation in each region is deduced. This step gives a detailed description of the stacking sequences by giving the arrangement of the plies in each region. Starting from the optimal configuration of the first step, the plies are permuted such that the panel comply with the blending and design rules and the buckling load is maximal.

This approach has the drawback that the sequences do not have a direct control over the thicknesses. If the second step gives a buckling load which is less than the safety threshold, then the thicknesses must be increased otherwise the overall optimization fails. If the second step gives a buckling load larger than the safety threshold, then the weight is not optimal and the thicknesses must be decreased. Moreover the manufacturing constraints are hardly satisfied using the penalty method. The computational cost is high because two optimization problems are solved with two expensive methods.

The proposed bilevel approach overcomes these drawbacks by seeking simultaneously the thicknesses and the stacking sequences. At each iteration, the buckling load is computed using the current admissible stacking sequences. The thicknesses are updated according to this computed value. Then, the stacking sequences are updated based on the new thickness values.

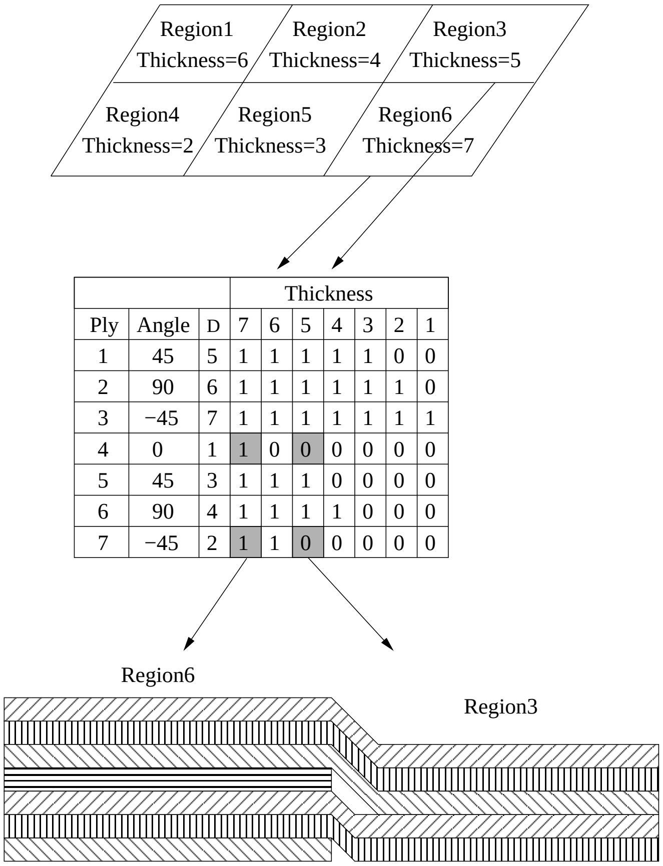
Paper [20] studies the optimization problem of the second step. It considers a structure with different regions. Each region has its own thickness and it is defined by the number of plies. These thicknesses can be defined from a preliminary design of the structure and they are the input of the optimization problem of our interest. The numerical example in the paper takes a panel of 8x6 regions, one region has 35 plies, some regions have 25 plies, others have 23 plies... The optimization problem seeks to maximize the buckling load with respect to the stacking sequence of each region such that the structure is manufacturable. A new optimization algorithm is proposed in [20]. It has the advantage over the existing ones (GA, topology optimization) that all the manufacturing constraints are satisfied at each iteration. Note that this second step, and thus the proposed optimization algorithm, does not affect the weight of the structure. It only improves its reserve factor given the thicknesses of the regions.

This paper proposes a new approach for the design of a composite structure which combines the two steps into one. The underlying optimization problem is the following. Minimize the weight of the structure with respect to the thicknesses of the regions such that the reserve factor is higher than a threshold and the stacking sequences are manufacturable. A new algorithm is proposed to solve this new formulation based on the one in [20]. Satisfying the constraint in the new formulation, which is the reserve factor is higher than a threshold and the stacking sequences are manufacturable, is the same as the optimization problem in [20]. For this reason, the new algorithm is based on this one. The new feature of this new algorithm with respect to the one in [20] is that it minimizes the weight and finds the thicknesses of the regions of the structure in addition to finding the corresponding manufacturable stacking sequences.

3. Parametrization of the Ply Drop-offs and Design Rules

Consider a panel composed of six regions, each one has its own thickness (see Figure 1). Let A and B be two adjacent regions such that the thickness of B is smaller or equal to the thickness of A. If the panel is blended, then the set of plies composing region B must be a subset of the one of region A: some plies of region A are prolonged into region B and the others are dropped.

The ply drop-offs are not fixed in advance but they are parametrized with a permutation vector D. It is explained in the following. Let N be the number of plies of the sequence guide. Consider a $N \times N$ lower triangular matrix. This matrix defines a ply drop-off scheme. Each column of this matrix gives the set of plies for a given thickness. The first column corresponds to the thickness equal to N plies and has all its rows equal to one. The second column corresponds to the thickness equal to $N-1$ and has all its rows equal to one except the first one which is equal to zero. This means that the region of thickness $N-1$ has all the plies of the region of thickness N except the first one. The third column has all its rows equal to one except the first two which are equal to zero. Thus, plies number 1 and 2 are dropped to get a stacking sequence of N-2 plies from a



			Thickness						
Ply	Angle	D	7	6	5	4	3	2	1
1	45	5	1	1	1	1	1	0	0
2	90	6	1	1	1	1	1	1	0
3	−45	7	1	1	1	1	1	1	1
4	0	1	1	0	0	0	0	0	0
5	45	3	1	1	1	0	0	0	0
6	90	4	1	1	1	1	0	0	0
7	−45	2	1	1	0	0	0	0	0

Figure 1: The blending principle and the ply drop-offs definition from a permutation vector D.

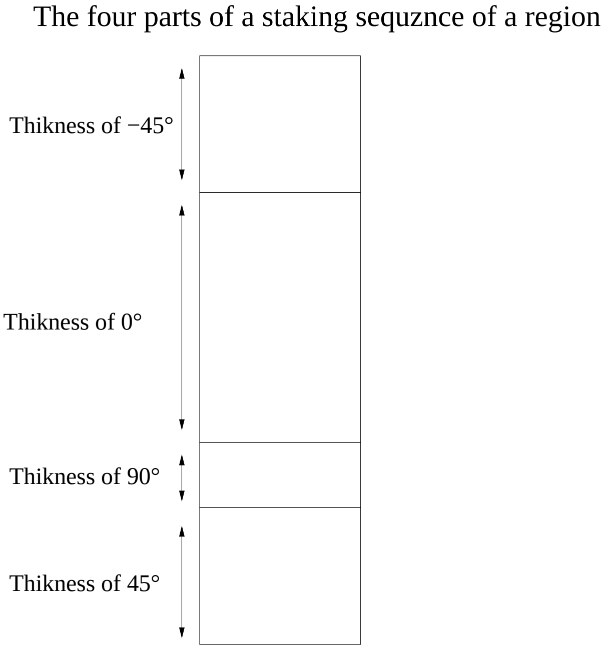

The four parts of a staking sequznce of a region

Figure 2: The pattern of the stacking sequence in step one.

sequence of N plies and ply number 2 is dropped to get a stacking sequence of N-2 plies from a sequence of N-1 plies. The same reasoning applies to any thickness between N and 1. This ply drop-off scheme defined by a lower triangular matrix is a blended one because the set of plies of a region of thickness is present in all the thicker regions.

Now if the rows of this lower triangular matrix are permuted according to a permutation vector D, another ply drop-off scheme which is also blended is defined because it is based on the lower triangular matrix. Therefore, the vector D can define all the possible ply drop-off schemes. This permuted lower triangular matrix is called a drop-off table. An element (i, j) of this table indicates whether ply i belongs to the set of plies of thickness j or not. Note that the number of regions in the panel is independent from the number of columns in the drop-off table. The stacking sequence of each region is deduced from its thickness and its corresponding column in the drop-off table. The table gives a stacking sequence for any value of the thickness even if no region in the panel has this value.

Figure 1 shows a drop-off table including this permuted lower matrix. In the example, $D = (5, 6, 7, 1, 3, 4, 2)$, region 6 of thickness 7 has the plies $(1, 2, 3, 4, 5, 6, 7)$ and region 3 of thickness 5 has of the plies $(1, 2, 3, 5, 6)$. The plies which are dropped between these two regions are grayed in the table. They correspond to the plies having entries equal to 1 in the thickness 7 and 0 in the thickness 5. Note that with 7 plies, it is possible to have 7 different values for the thickness, even though only 6 of them are present in this particular panel.

Let u^i be the ply sequence of the region i of the panel. We have $u^1 = (1, 2, 3, 5, 6, 7)$, $u^2 = (1, 2, 3, 6)$, $u^3 = (1, 2, 3, 5, 6)$, $u^4 = (2, 3)$ $u^5 = (1, 2, 3)$ and $u^6 = (1, 2, 3, 4, 5, 6, 7)$. The design rules have to be applied to these ply sequences: the ply orientations of the stacking sequence guide must be chosen such that these ply sequences are admissible. The design rules are the following.

- The orientation in each ply must be chosen such that two consecutive plies do not have a gap in the orientation equal to 90^o. Thus, $(0^o, 90^o)$ and $(-45^o, 45^o)$ cannot be two consecutive plies in all the regions of the panel. This rule aims to reduce the delamination risk at free edges.

- Maximum four consecutive plies in each region can have the same orientation. This rule reduces the progressive damage between the plies.

- Symmetric ply sequences. The laminate is symmetric so the stiffness matrix B is equal to zero. In this case, there is no coupling between the in-plane and out-of-plane effects. This also avoids bending and torsion of the laminate during the cooling phase which follows the curing process.

- A fixed number of plies of each orientation is defined in each region. This constraint is typically found in a global/local optimization framework as it is explained in [22, 23] or with the two-steps optimization method in section 2 and [14]. Note that this rule includes the case of a balanced sequence.

Other constraints, can also be considered. For example, avoiding the sandwich laminates by imposing an additional manufacturing rule which is imposing a maximum of four consecutive dropped plies between two adjacent regions. In the following section, the mathematical formulation of the design rules is derived. It gives the mathematical relationship between the angles sequences and the ply drop-offs.

4. Mathematical Formulation of Design Rules

Each ply can have one of the four possible orientations which are $(-45^o, 0^o, 45^o, 90^o)$. For each ply $i = 1..N$, there are four binary variables $S_i^{-45}, S_i^0, S_i^{45}, S_i^{90}$ which represent the presence or the absence of orientation θ at ply i. Let $S = \{S^{-45}, S^0, S^{45}, S^{90}\} \in \{0,1\}^{4N}$ be a vector which represents the orientation sequence of the thickest region. S is the stacking sequence guide. One can deduce the first constraint on S which says that only one orientation is assigned to each ply:

$$S_i^{-45} + S_i^0 + S_i^{45} + S_i^{90} = 1, \quad i = 1..N.$$

Let N_r be the number of regions in the panel, $t \in \mathbb{N}^{N_r}$ the vector of thicknesses of all the regions and u^r the ply sequence of the region r. In the previous example we had $N_r = 6$, the thicknesses of the regions t^r were $\{6, 4, 5, 3, 5, 7\}$ and the ply sequence u^r of each region r is given by $u^1 = (1, 2, 3, 5, 6, 7)$, $u^2 = (1, 2, 3, 6)$, $u^3 = (1, 2, 3, 5, 6)$, $u^4 = (2, 3)$, $u^5 = (1, 2, 3)$ and $u^6 = (1, 2, 3, 4, 5, 6, 7)$. The design rules must be applied to each of these sequences. Note that these sequences are derived from the specific ply drop-off scheme in the example, for which $D = (5, 6, 7, 1, 4, 2)$.

The rule which prevents two consecutive plies from having an orientation gap of 90^o is written as:

$$\begin{array}{rcl}
S_{u_i^r}^{-45} + S_{u_{i+1}^r}^{45} & < & 2 \\
S_{u_i^r}^{0} + S_{u_{i+1}^r}^{90} & < & 2 \\
S_{u_i^r}^{45} + S_{u_{i+1}^r}^{-45} & < & 2 \\
S_{u_i^r}^{90} + S_{u_{i+1}^r}^{0} & < & 2, \quad r = 1..N_r,\ i = 0..t^r - 1,
\end{array}$$

where u_i^r and u_{i+1}^r are the indexes of two consecutive plies in a region r.

The rule of having maximum four consecutive identical angles in each region is expressed as:

$$\begin{array}{rcl}
S_{u_i^r}^{-45} + S_{u_{i+1}^r}^{-45} + S_{u_{i+2}^r}^{-45} + S_{u_{i+3}^r}^{-45} + S_{u_{i+4}^r}^{-45} & < & 5 \\
S_{u_i^r}^{0} + S_{u_{i+1}^r}^{0} + S_{u_{i+2}^r}^{0} + S_{u_{i+3}^r}^{0} + S_{u_{i+4}^r}^{0} & < & 5 \\
S_{u_i^r}^{45} + S_{u_{i+1}^r}^{45} + S_{u_{i+2}^r}^{45} + S_{u_{i+3}^r}^{45} + S_{u_{i+4}^r}^{45} & < & 5 \\
S_{u_i^r}^{90} + S_{u_{i+1}^r}^{90} + S_{u_{i+2}^r}^{90} + S_{u_{i+3}^r}^{90} + S_{u_{i+4}^r}^{90} & < & 5, \quad r = 1..N_r,\ i = 0..t^r - 1.
\end{array}$$

The rule of having a fixed number of plies of each orientation is expressed as:

$$\begin{aligned}
\sum_{i=0}^{t^r-1} S^{-45}_{u^r_i} &= n^r_{-45},\\
\sum_{i=0}^{t^r-1} S^{0}_{u^r_i} &= n^r_{0},\\
\sum_{i=0}^{t^r-1} S^{45}_{u^r_i} &= n^r_{45},\\
\sum_{i=0}^{t^r-1} S^{90}_{u^r_i} &= n^r_{90}, \quad r = 1..N_r
\end{aligned}$$

where n^r_θ is the fixed number of plies having orientation θ in the region r. Taking $n^r_{45} = n^r_{-45}$ makes the structure balanced.

Finally with the symmetry constraint, only the half upper part of the stacking sequence guide is considered. This constraint is coupled with the constraint of having maximum four identical plies. The orientations of the last three plies from the middle of the sequence must not be identical otherwise we have six consecutive identical plies:

$$\begin{aligned}
S^{-45}_{u^r_{t^r}} + S^{-45}_{u^r_{t^r-1}} + S^{-45}_{u^r_{t^r-2}} &< 3\\
S^{0}_{u^r_{t^r}} + S^{0}_{u^r_{t^r-1}} + S^{0}_{u^r_{t^r-2}} &< 3\\
S^{45}_{u^r_{t^r}} + S^{45}_{u^r_{t^r-1}} + S^{45}_{u^r_{t^r-2}} &< 3\\
S^{90}_{u^r_{t^r}} + S^{90}_{u^r_{t^r-1}} + S^{90}_{u^r_{t^r-2}} &< 3 \quad r = 1..N_r.
\end{aligned}$$

All of these constraints are linear combinations of the elements of S and they depend on D through the indexes u^r_i. These constraints can be written in a condensed matrix form as:

$$C_t(D)S \leqq z, \tag{1}$$

where $C_t(D)$ is a binary matrix and z is a vector grouping right-hand sides of the constraints equations.

5. The ingredients of the Optimization Algorithm

In this section, the ingredients of the optimization algorithm are given.

5.1. The Levenshtein Distance Between Two Sequences

The Levenshtein distance between two stacking sequences is the number of operations needed to transform one sequence into another one. The allowed operations are insert, delete and change. For example, let $S_1 = \{-45, 0, 45, 90\}$ and $S_2 = \{-45, 0, 45, 0, 90\}$. The Levenshtein distance between these two sequences

is equal to one because one insertion of 0 is needed to transform S_1 into S_2. This distance measures the amount of difference between two sequences. It is a metric that is used to define later a neighborhood of a sequence.
This definition is extended to the distance between two panels defined be (t_1, S_1, D_1) and (t_2, S_2, D_2). This distance, $\mathscr{D}((t_1, S_1, D_1), (t_2, S_2, D_2))$, is the sum over the regions of the distances between the respective sequences.

5.2. The Primal and the Dual Problems

First, the following primal problem is defined. for a given ply drop-off scheme D and a vector t of N_r thicknesses , find all sequence guides S which are admissible with respect to D and t. Let $\mathcal{A}_t(D)$ be this set of admissible sequence guides:

$$\mathcal{A}_t(D) = \big\{S, \text{ such that } C_t(D)S \leqq z \text{ and } S \in \{0,1\}^{4N}\big\}. \tag{2}$$

Second, the following dual problem is defined. For a given sequence guide S and thickness vector t, find all the ply drop-offs D for which S is admissible. In this case, $\mathcal{A}'_t(S)$ denotes the dual set:

$$\mathcal{A}'_t(S) = \big\{D, \text{ such that } C_t(D)S \leqq z \text{ and } D \in \mathrm{Sym}(1..N)\big\}, \tag{3}$$

where Sym$(1..N)$ is the group of all permutations over the set of integers $(1..N)$. The enumeration of the elements of these sets is based on backtracking algorithms which will be presented in Section 6.
Let $\mathscr{D}(S_1, S_2)$ be the Levenshtein distance between two sequences and $\mathscr{D}(D_1, D2)$ be the one between two ply drop-offs. We finally define the neighborhood of a sequence $S_1 \in \mathcal{A}_t(D)$ and the one of a ply drop-off $D_1 \in \mathcal{A'}_t(S)$ as:

$$\begin{aligned} \mathcal{V}^{d_0}_{D,t}(S_1) &= \{S_2 \in \mathcal{A}_t(D) \text{ such that } \mathscr{D}(S_1, S_2) \leq d_0\} && (4) \\ \mathcal{V}^{d_0}_{D,t}(D_1) &= \{D_2 \in \mathcal{A}'_t(S) \text{ such that } \mathscr{D}(D_1, D_2) \leq d_0\}, && (5) \end{aligned}$$

for some predefined integer d_0. The first one is the set of admissible sequence guides with respect to D that differ from S_1 by d_0 operations. The second one is the set of ply drop-offs for which S is admissible and differ from D_1 by d_0 operations.

5.3. The Projection Operator

Let $\mathcal{M}(t)$ be the set of couples (S, D) such that $S \in \mathcal{A}_t(D)$. It is the set of (S, D) such that the panel has thickness vector t and the sequence of each region is admissible. The projection operator is explained using the example in Figure 3. The first panel has the thickness vector $t_1 = \{5, 3, 2\}$ and the stacking sequences are defined by (S_1, D_1) in the figure. The second panel has the thickness vector $t_2 = \{6, 4, 1\}$ but its stacking sequences are defined by (S_2, D_2) are unknown. One can see that the thicknesses of the first panel do not fit the second panel. One ply has to be added to regions 1 and 2 respectively and one ply has to be

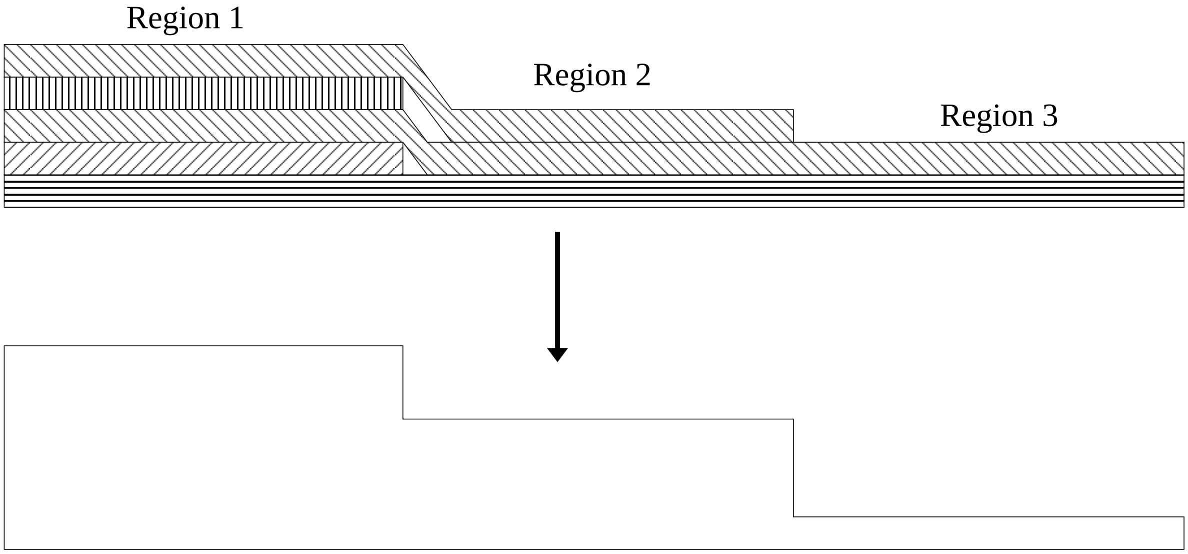


Figure 3: The projection operator: finding the stacking sequence guide and the ply drop-offs of a panel which are the most similar to a reference ones.

removed from region 3. The projection operator seeks to find $(S_2, D_2) \in \mathcal{M}(t_2)$ such that the Levenshtein distance between these two panels is minimal. It gives the panel of thickness vector t_2 such that its stacking sequences are admissible and they are the most similar to the ones of the first panel. The projection operator is formulated as the following optimization problem:

$$\begin{aligned} &\min_{(S_2,D_2)} \mathscr{D}((t_1, S_1, D_1), (t_2, S_2, D_2)) \\ &\text{s.t. } (S_2, D_2) \in \mathcal{M}(t_2) \end{aligned} \tag{6}$$

6. Constraint Satisfaction with a Backtracking Algorithm

In this section, the backtracking algorithm is presented. It aims at solving the primal and dual problems defined in (2 and 3) with a combinatorial procedure. Only the general idea of algorithm is presented. A detailed explanation of the backtracking concept can be found in [24, 25, 26].

Finding admissible sequence guides or ply drop-offs is not a trivial task given the combinatorial nature of the constraints. Most of the time, one cannot guess intuitively such sequences and computer-based algorithms must be used to perform this task.

The easiest but not the most efficient way to find sequence guides which are admissible for a given ply drop-off is the so-called brute-force enumeration. It consists in enumerating all the sequence possibilities and checking for each one its admissibility. The main disadvantage of this method is that its computational cost grows exponentially with the number of plies. For example, for $N = 16$ plies there are $4^{16} = 4294967296$ possibilities to be checked and for $N = 32$ plies there

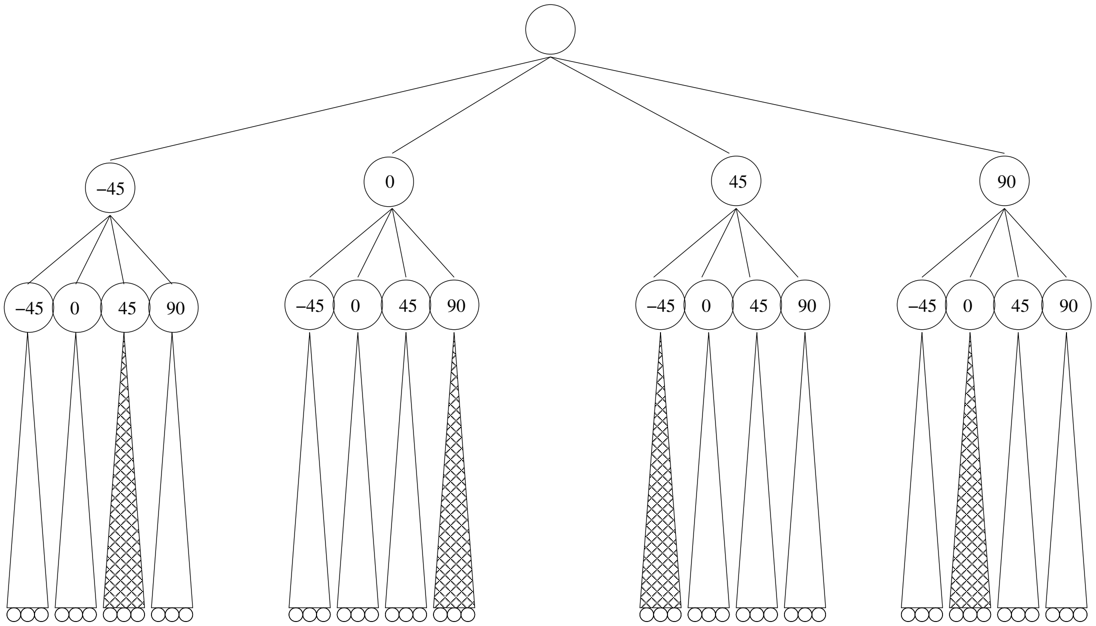


Figure 4: Backtracking enumeration tree with pruned branches

are $4^{32} \approxeq 1.844 \times 10^{19}$ possibilities! A more sophisticated technique has to be used in order to decrease the number of possibilities to be checked.

Enumerating all possible sequence guides consists in building an enumeration tree like in Figure 4. Each level of the tree represents a ply and each node has four children which are the four possible angle values of the next ply. The enumeration tree must have $N + 1$ levels. A stacking sequence guide is a branch of the tree connecting the root to a leaf (the lowest node). One can see that the size of the tree grows exponentially with N and spanning the whole tree becomes quickly unfeasible.

The idea of the backtracking is to span the entire tree and to check at each node the admissibility of the partial stacking sequence constituted by the branch going from the root to the current node. If the partial sequence violates the constraint, then all the sub tree derived from the current node is eliminated (see Figure 4). This pruning technique reduces considerably the size of the tree and makes the enumeration efficient. For example in Figure 4, all the sub-sequences starting with $(-45, 45)$, $(0, 90)$,$(45, -45)$ and $(90, 0)$ are pruned from the tree because they violate the 90^o gap rule. The leaves of the pruned tree are only the admissible sequence guides.

7. The Numerical Algorithm

The optimization problem is formulated as follows:

$$\begin{aligned} &\min_{t} \mathscr{W}(t) \\ &\text{s.t.} \quad \max_{(S,D)} \mathscr{B}(t, S, D) > c, \\ &\qquad (S, D) \in \mathcal{M}(t), \end{aligned} \tag{7}$$

where $\mathscr{W}(t)$ is the weight of the structure depending on the thickness vector t, $\mathscr{B}(t, S, D)$ is the buckling load and it must be higher that a safety threshold c. The couple (S, D) must give in each region an admissible sequence with respect to the blending and the manufacturing rules. Note that the proposed optimization method is not restricted to a certain type of a reserve factor. One can choose buckling, delamination, micro strains, reparability or any other reserve factor.

The objective function only depends on t. For each value of t, a couple (S, D) is sought such that the buckling load $\max \mathscr{B}(t, S, D)$ is higher than c and in the admissible set of sequences defined by $\mathcal{M}(t)$. Seeking such a couple (S, D) is formulated as an optimization subproblem and the over whole formulation is a bilevel one.

The optimization algorithm is the following. At iteration k, (S_k, D_k) are in $\mathcal{M}(t_k)$. The buckling load $\mathscr{B}(t_k, S_k, D_k)$ is evaluated and the update of t_k is based on the value of the buckling load. If $\mathscr{B}(t_k, S_k, D_k) > c$, t_k is updated with a random value vector t_{k+1} such that it is in the neighborhood of t_k and $\mathscr{W}(t_{k+1}) < \mathscr{W}(t_k)$. If $\mathscr{B}(t_k, S_k, D_k) < c$, t_k is updated with a random value vector t_{k+1} such that it is in the neighborhood of t_k and $\mathscr{W}(t_{k+1}) > \mathscr{W}(t_k)$.

The update of t_k is directly connected to the value of $\mathscr{B}(t_k, S_k, D_k)$. Given t_{k+1}, (S_k, D_k) is updated as follows. It is projected over $\mathcal{M}(t_{k+1})$ to find a $(S_k, D_k)^p$ which fits the new thickness vector t_{k+1}. The optimization subproblem of seeking $\mathscr{B} > c$ is solved by taking $(S_k, D_k)^p$ as an initial point and (S_{k+1}, D_{k+1}) is set to the solution of the optimization subproblem.

The optimization algorithm requires, at each iteration k, solving two optimization subproblems: the projection operator and maximizing the buckling load. However, these two subproblems are not expensive in computations, because the first one is based on an analytical function (the Levenshtein distance) and the second one takes an initial point the projection of the solution of the previous iteration. The optimization algorithm is summarized in Algorithm 1.

Algorithm 1 Optimization Algorithm

```
Define $t_0$ the initial thickness vector, the initial sequence guide and the ply drop-offs $(S_0, D_0) \in \mathcal{M}(t_0)$.
for $k = 0$ to maximal number of iterations do
    Compute $\mathscr{B}(t_k, S_k, D_k)$ and $\mathscr{W}(t_k)$.
    if $\mathscr{B}(t_k, S_k, D_k) > c$ then
        Generate a random $t_{k+1}$ such that $\mathscr{W}(t_{k+1}) < \mathscr{W}(t_k)$ and it is in the neighborhood of $t_k$.
    else
        Generate a random $t_{k+1}$ such that $\mathscr{W}(t_{k+1}) > \mathscr{W}(t_k)$ and it is in the neighborhood of $t_k$.
    end if
    Compute $(S_k, D_k)^p$ the projection of $(S_k, D_k)$ over $\mathcal{M}(t_{k+1})$.
    Optimize $\mathscr{B}$ by taking $(S_k, D_k)^p$ as an initial point.
    Set $(S_{k+1}, D_{k+1})$ to the solution of the previous step.
end for
```

The algorithm for solving these optimization subproblems is in the following section.

Note that, the proposed algorithm is a local optimization method. It only finds local minima and does not converge to the global one. This remark is added to the text.

7.1. Solving the Optimization Subproblem

For a given thickness vector t, the optimization subproblem is formulated as follows:

$$\begin{aligned} &\max_{S,D} F(t, S, D) \\ &\text{s.t.} \quad (S, D) \in \mathcal{M}(t), \end{aligned} \tag{8}$$

where F is either the distance function or the buckling load function.

At each iteration k, (S_k, D_k) are updated to (S_{k+1}, D_{k+1}) such that S_{k+1} is admissible with respect to D_{k+1}. This task is performed in two steps (see Figure 5). First D_k is fixed and a random sequence guide candidate S_c is generated in $\mathcal{V}^{d_0}_{D_k}(S_k)$, the neighborhood of S_k which comply with D_k. Then, S_c is fixed and a random ply drop-off candidate D_c is generated in $\mathcal{V}^{d_0}_{S_c}(D_k)$, the neighborhood of D_k for which S_c is admissible. The buckling load $\mathscr{B}(t, S_c, D_c)$ is computed. If $\mathscr{B}(t, S_c, D_c) > \mathscr{B}(t, S_k, D_k)$ then $(S_{k+1}, D_{k+1}) = (S_c, D_c)$, otherwise $(S_{k+1}, D_{k+1}) = (S_k, D_k)$.

The operation of generating random elements in the neighborhood of the current iteration is called local search and it is based on the backtracking procedure described in the previous section. For more details about this optimization technique see [25].

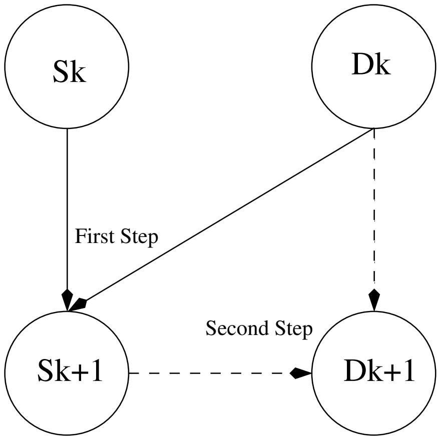


Figure 5: (S_k, D_k) are updated to (S_{k+1}, D_{k+1}) in two steps: first S_{k+1} from (S_k, D_k), second D_{k+1} from (S_{k+1}, D_k).

This is summarized in Algorithm 2.

Algorithm 2 Algorithm of the optimization subproblem

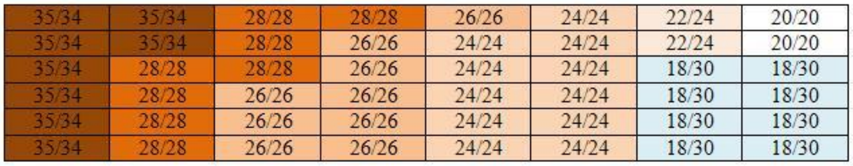
```
Define a neighborhood $d_0$, a permutation vector $D_0 = (0..N-1)$, a stacking
sequence guide $S_0$ in $\mathcal{A}(D_0)$ and $k = 0$
for $k = 0$ to maximal number of iterations do
  Generate a random sequence $S_c \in \mathcal{V}^{d_0}_{D_k}(S_k)$.
  Generate a random ply drop-offs $D_c \in \mathcal{V}^{d_0}_{S_c}(D_k)$.
  Compute $F(S_c, D_c)$.
  if $F(S_c, D_c) > F(S_k, D_k)$ then
    $(S_{k+1}, D_{k+1}) = (S_c, D_c)$
  else
    $(S_{k+1}, D_{k+1}) = (S_k, D_k)$
  end if
end for
```

8. Numerical experiments

Some numerical experiments are carried out to illustrate the performance of the algorithm. First, the projection operator is studied to show its capability to find the most similar stacking sequence guide and the ply drop-offs in a set of admissible ones. Next, the overall optimization is performed with the proposed algorithm.

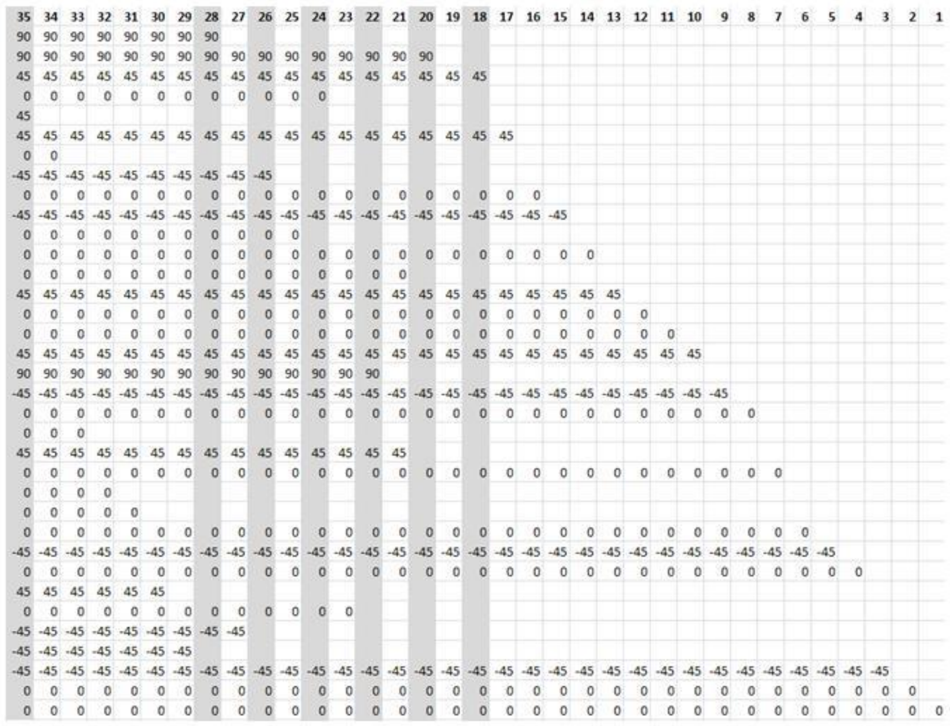
Test case 1: the projection operator. Consider a panel P_1 with defined thicknesses and stacking sequences and a panel P_2 with only its thicknesses known. The thicknesses of the regions in panel P_1 and panel P_2 are expressed in number of plies and are shown in Figure 6. This figure shows one panel representing panels

P_1 and P_2. Each region of contains the number of plies of P_1 and P_2 respectively. They are denoted by the vectors t_1 and t_2. The sequences of the regions of panel P_1 are defined by the couple (S_1, D_1) in Figure 7. This figure is the drop-off table which corresponds to panel P_1 in Figure 6. As it is shown in Figure 6, panel P_1 has regions with 35 plies, 28 plies, 26 plies, 24 plies, 22 plies, 20 plies and 18 plies. Each region has its own stacking sequence given in Figure 7. Like the drop-off table presented in section 3 and Figure 1, each column represents a stacking sequence. The first row of the column is the number of plies and then the stacking itself. The grayed columns in Figure 7 correspond to the thicknesses of our interest because they are ones in panel P_1. The same reasoning applies to Figure 9. The percentages of 20% of -45^o, 50% of 0^o, 20% of 45^o and 10% of 90^o are imposed to the number of plies in each region.



35/34	35/34	28/28	28/28	26/26	24/24	22/24	20/20
35/34	35/34	28/28	26/26	24/24	24/24	22/24	20/20
35/34	28/28	28/28	26/26	24/24	24/24	18/30	18/30
35/34	28/28	26/26	26/26	24/24	24/24	18/30	18/30
35/34	28/28	26/26	26/26	24/24	24/24	18/30	18/30
35/34	28/28	26/26	26/26	24/24	24/24	18/30	18/30

Figure 6: The thicknesses of the two panels used for testing the projection operator. This figure shows one panel representing panels P_1 and P_2. Each region of contains the number of plies of P_1 and P_2 respectively.



35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
90	90	90	90	90	90	90	90																											
90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90																			
45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45																	
0	0	0	0	0	0	0	0	0	0	0	0																							
45																																		
45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45																
0	0																																	
-45	-45	-45	-45	-45	-45	-45	-45	-45	-45																									
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0															
-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45														
0	0	0	0	0	0	0	0	0	0	0																								
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0													
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																				
45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45												
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0											
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0										
45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45									
90	90	90	90	90	90	90	90	90	90	90	90	90	90																					
-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45								
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							
0	0	0																																
45	45	45	45	45	45	45	45	45	45	45	45	45	45	45																				
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						
0	0	0	0																															
0	0	0	0	0																														
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45				
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
45	45	45	45	45	45																													
0	0	0	0	0	0	0	0	0	0	0	0	0																						
-45	-45	-45	-45	-45	-45	-45	-45	-45																										
-45	-45	-45	-45	-45	-45	-45																												
-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 7: Test of the projection operator, the stacking sequences of the first panel.

The first test with the projection operator is a validation test which consists in projecting (S_1, D_1) over $\mathcal{M}(t_1)$. The projection of (S_1, D_1) over $\mathcal{M}(t_1)$ is the same couple (S_1, D_1) given that $(S_1, D_1) \in \mathcal{M}(t_1)$. The minimal distance is equal to zero. It is a trivial test but interesting because the solution is known in advance. The evolution of the distance with respect to the iteration number is shown in Figure 8. This test shows that the proposed optimization method described in the subsection 7.1 is able to find the exact solution of the optimization subproblem.

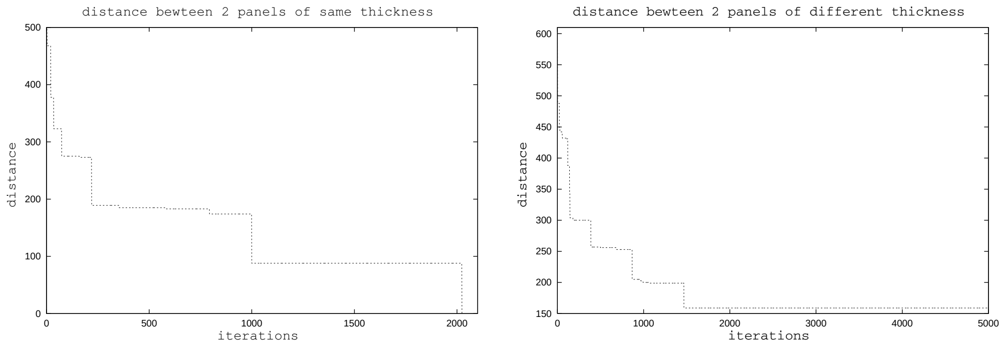
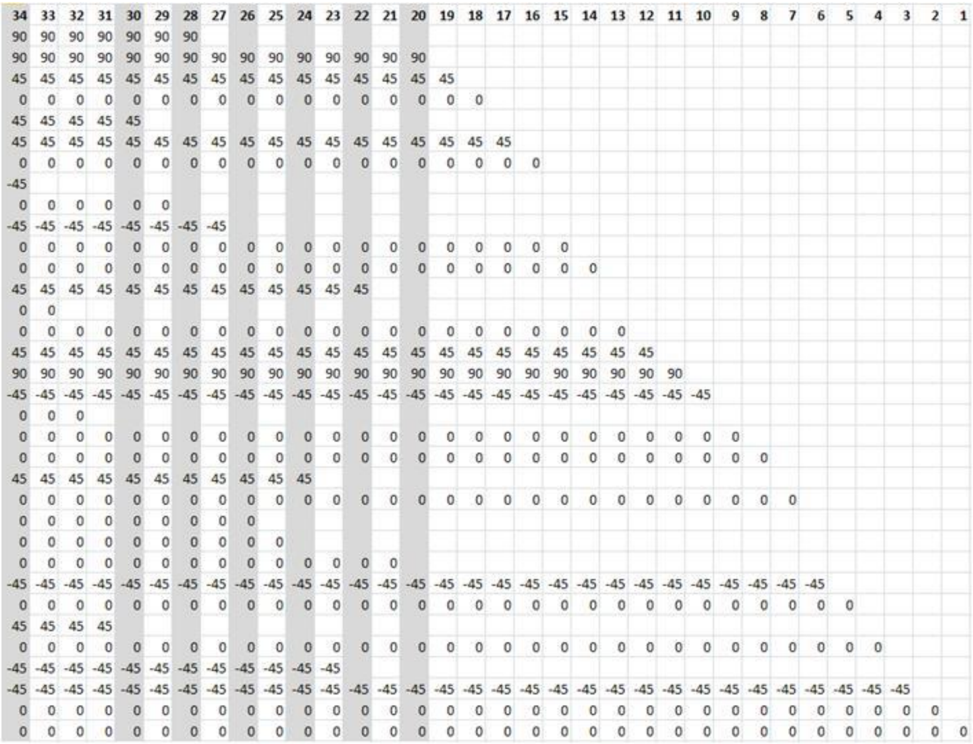


Figure 8: The evolution of the distance with respect to the number of iterations for tests one (left) and two (right).



34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
90	90	90	90	90	90	90																											
90	90	90	90	90	90	90	90	90	90	90	90	90	90	90																			
45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45																		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																	
45	45	45	45	45																													
45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45																
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0															
-45																																	
0	0	0	0	0	0																												
-45	-45	-45	-45	-45	-45	-45	-45																										
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0														
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0													
45	45	45	45	45	45	45	45	45	45	45	45	45																					
0	0																																
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0												
45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45											
90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90										
-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45									
0	0	0																															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							
45	45	45	45	45	45	45	45	45	45	45																							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						
0	0	0	0	0	0	0	0	0																									
0	0	0	0	0	0	0	0	0	0																								
0	0	0	0	0	0	0	0	0	0	0	0	0	0																				
-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45					
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
45	45	45	45																														
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45																						
-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 9: Test of the projection operator, the obtained stacking sequences of the second panel after projection.

The second case is the projection of (S_1, D_1) over $\mathcal{M}(t_2)$. Here, the solution is not known in advance. A minimal distance is found equal to 150. The evolution of the distance with the iteration number is shown in at the left of Figure 8 and the resulting stacking sequences are in Figure 9. One can see the similarity of the sequences of the two panels which demonstrates the efficiency of the projection operator.

Test case 2: the full panel optimization. The experiments are based on a panel divided into 8×6 regions. The panel has dimensions of $800mm \times 600mm$. It is simply supported and it is loaded in compression only along the x-axis with a charge equal to 1N. It is assumed that the composite panel is made of carbon material with a Young's modulus equal to $[13 \times 10^6, 4650, 4650] N.mm^{-2}$, a shear modulus equal to $[4650, 4650, 4650] N.mm^{-2}$ and a Poisson ratio equal to $[0.35, 0.35, 0.35]$. The ply thickness is equal to $0.125mm$. The finite elements analysis is run over a quadrangular mesh of size 40×30 with first order elements. Figure 10 shows the first three buckling modes of the panel computed using SAMCEF software. The first one is used as the buckling load for the optimization. The purpose of Figure 10 is to show the deformation of the panel under compression. It is just to have an idea about the behavior of the structure which is considered for optimization, but it is not related to the optimization itself.

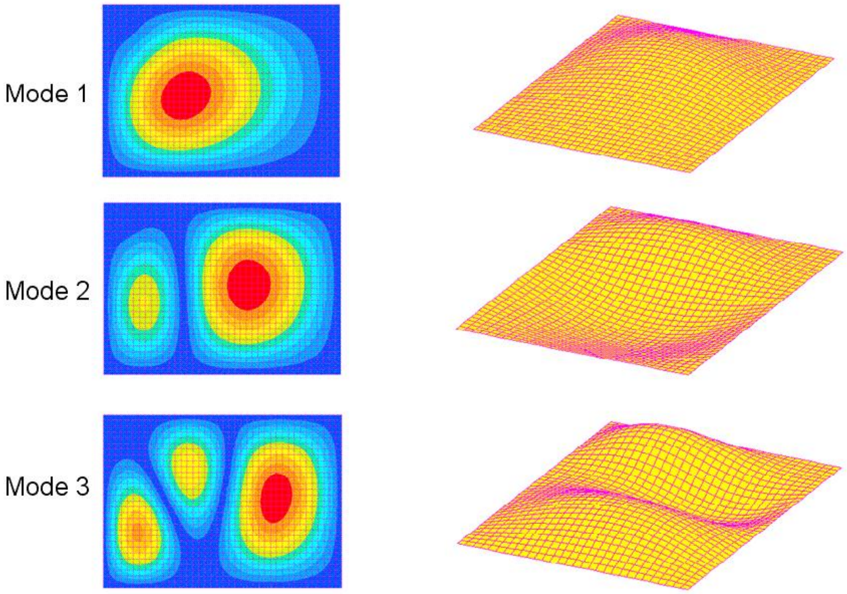


Figure 10: Test case 2, the first three buckling modes of the panel.

The region thicknesses of the panel are defined using a shape function. It is the bilinear function which interpolates the values of the thickness at the four corners of the panel. Thus, the vector t is of dimension four. The purpose of this choice is to reduce the degrees of freedom of the thicknesses and hence the computational cost. It is also to have a smooth distribution of the thickness and hence a manufacturable one.

The safety threshold of the buckling load c is equal to 500. This choice of this value is not based on a physical consideration. Given that the buckling load of the initial point before optimization is about 1400, the value 500 makes the in ital far from satisfying the constraint. Therefore, this value for c would illustrate the optimization of the structure. The update of t is as follows and it is based on the correlation between weight and buckling load. If $\mathscr{B} > c$, then one of the four coordinates of t is randomly decreased by one which makes the weight of the panel decrease. If $\mathscr{B} < c$ then one of the four coordinates of t is increased randomly by one which makes the weight of the panel increase.

Maximum 10 evaluations of $\mathscr{B}$ are allowed to solve the optimization subproblem. Two tests are performed: one with a fixed percentage of orientations (20% of -45^o, 50% of 0^o, 20% of 45^o and 10% of 90^o) in each region and one without this constraint.

Figures 11 and 12 show the evolution of the weight and the buckling load for the test case with the proportions constraint and Figures 13 and 14 are for the other test case. Table 1 gives the optimal values of the weight, the buckling loads and the thickness vector for both cases. Figures 15 and 16 show the sequences of the two test cases respectively. The grayed columns are the ones that correspond to the obtained thicknesses by optimization.

Test case	Weight	Reserve Factor	Thickness vector
without proportions constraints	2595	499	$(24, 21, 23, 19)$
with proportions constraints	2600	517	$(24, 21, 22, 20)$

Table 1: Results of test case 2: the full optimization.

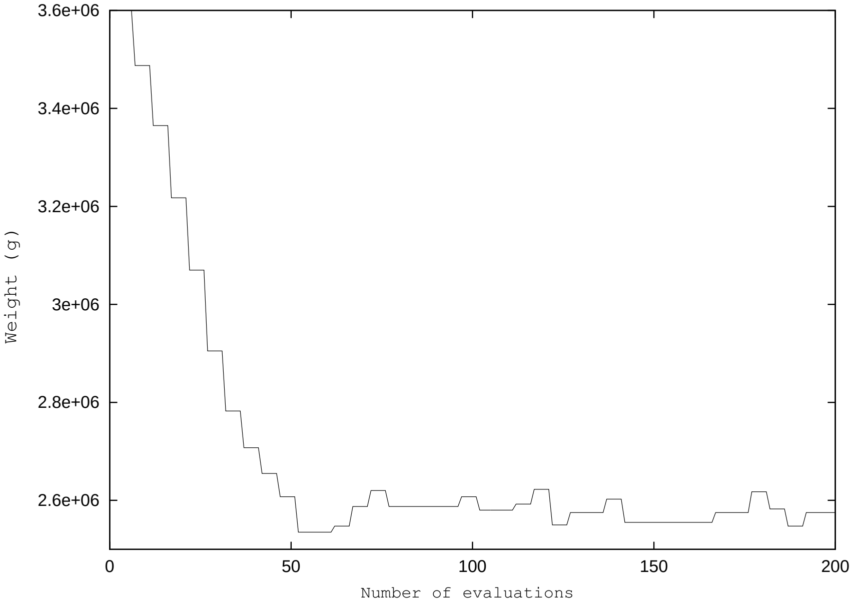


Figure 11: Test case 1 (without proportions constraint): evolution of the weight of the structure with respect to the evaluation number.

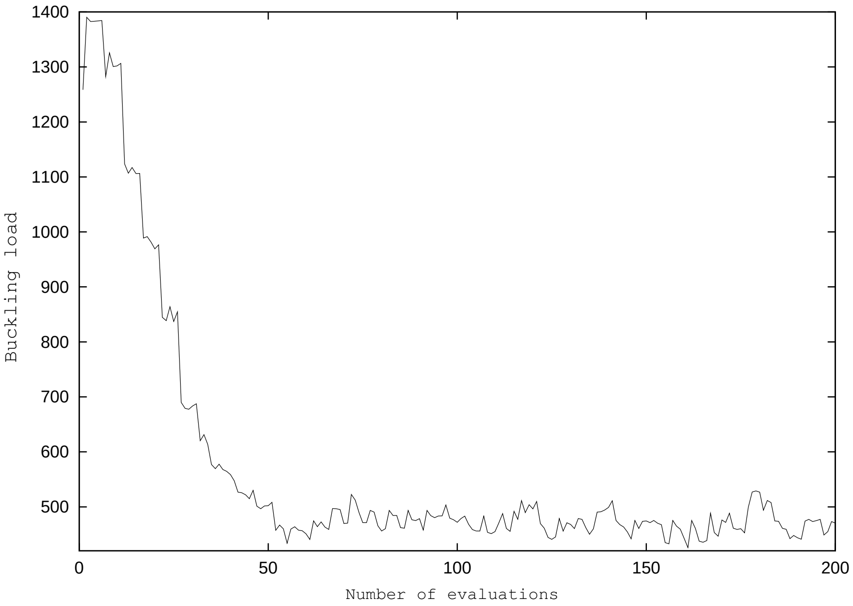


Figure 12: Test case 1 (without proportions constraint): evolution of the buckling load of the structure with respect to the evaluation number.

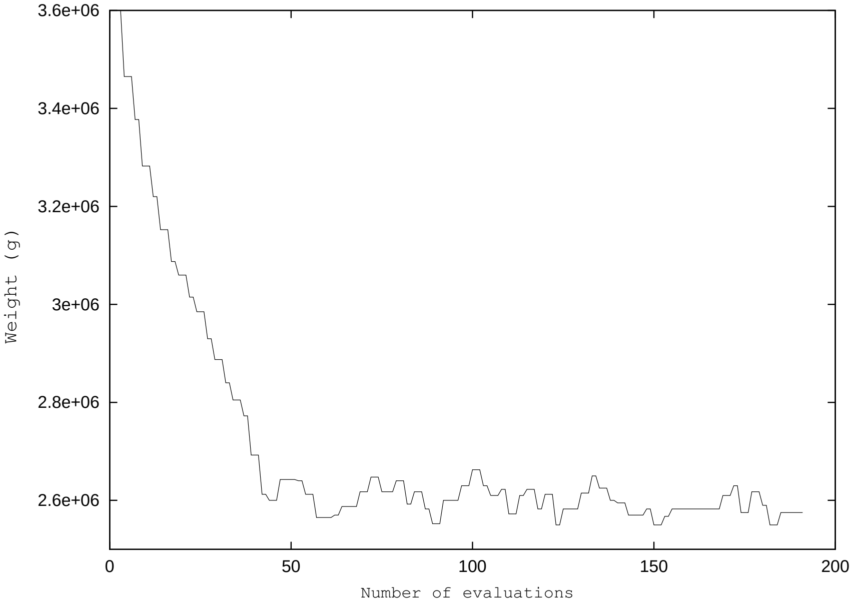


Figure 13: Test case 2 (with proportions constraint): evolution of the weight of the structure with respect to the evaluation number.

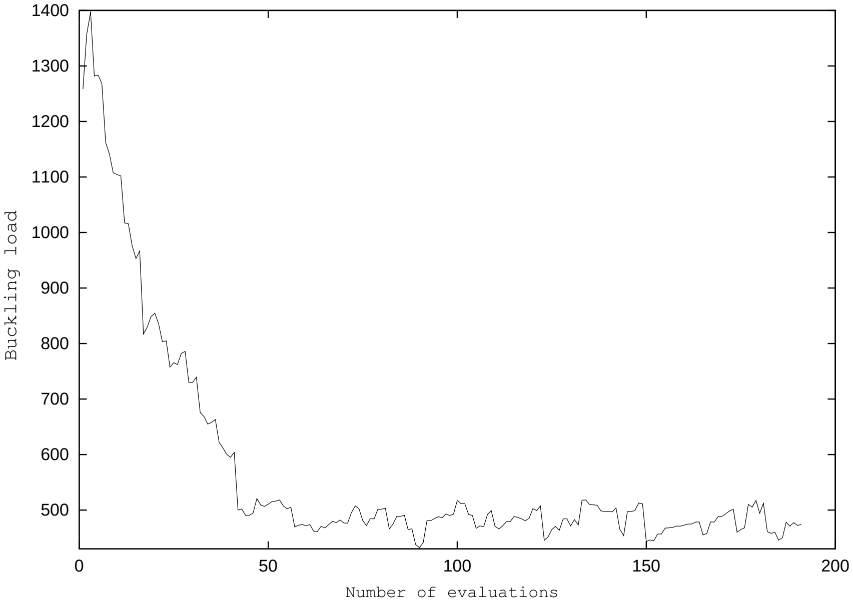
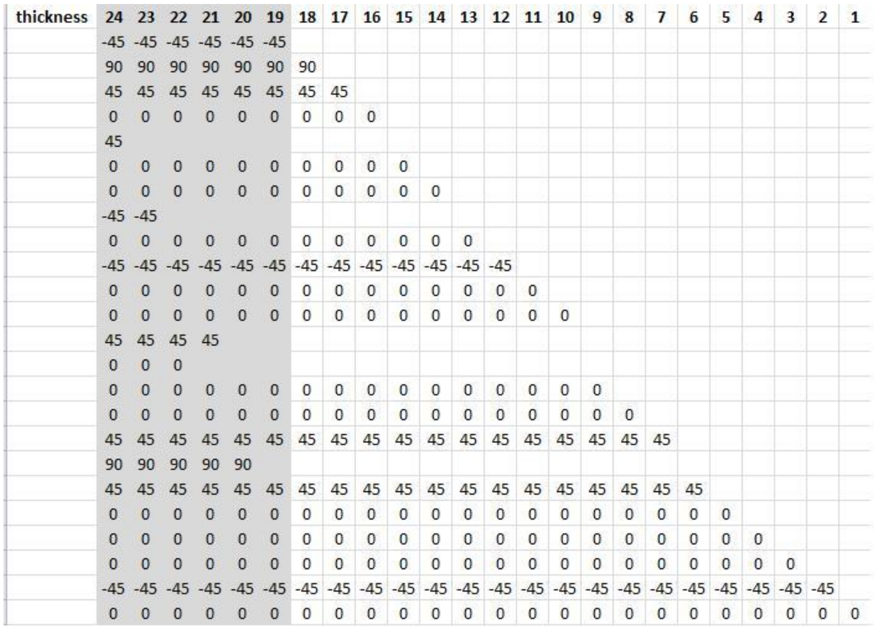
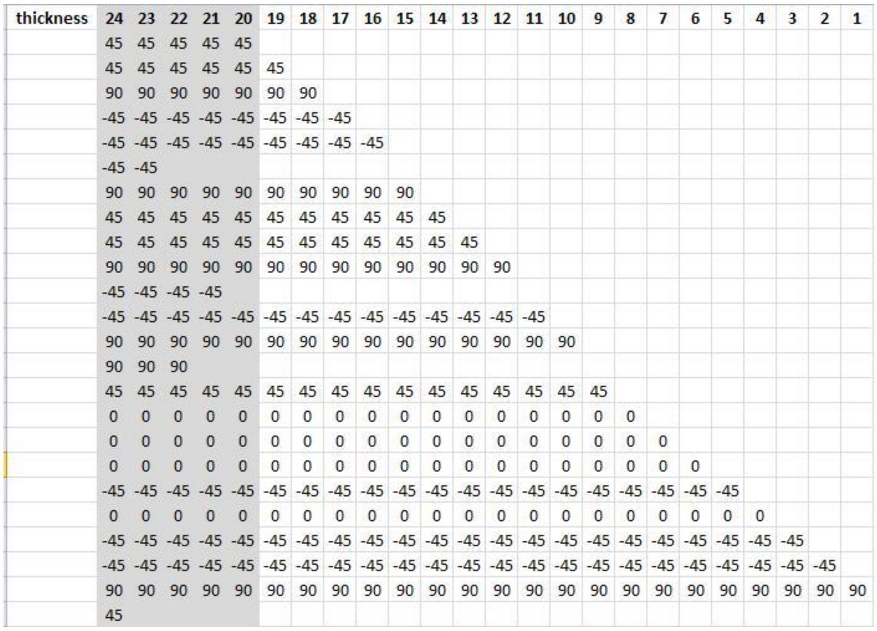


Figure 14: Test case 2 (with proportions constraint): evolution of the buckling load of the structure with respect to the evaluation number.



thickness	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	-45	-45	-45	-45	-45	-45																		
	90	90	90	90	90	90	90																	
	45	45	45	45	45	45	45	45																
	0	0	0	0	0	0	0	0	0															
	45																							
	0	0	0	0	0	0	0	0	0	0														
	0	0	0	0	0	0	0	0	0	0	0													
	-45	-45																						
	0	0	0	0	0	0	0	0	0	0	0	0												
	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45											
	0	0	0	0	0	0	0	0	0	0	0	0	0	0										
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0									
	45	45	45	45																				
	0	0	0																					
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							
	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45						
	90	90	90	90	90																			
	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45					
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 15: Test case 2 (without proportions constraint): the upper part of the symmetric stacking sequences.



thickness	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	45	45	45	45	45																			
	45	45	45	45	45	45																		
	90	90	90	90	90	90	90																	
	-45	-45	-45	-45	-45	-45	-45	-45																
	-45	-45	-45	-45	-45	-45	-45	-45	-45															
	-45	-45																						
	90	90	90	90	90	90	90	90	90	90														
	45	45	45	45	45	45	45	45	45	45	45													
	45	45	45	45	45	45	45	45	45	45	45	45												
	90	90	90	90	90	90	90	90	90	90	90	90	90											
	-45	-45	-45	-45																				
	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45										
	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90									
	90	90	90																					
	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45	45								
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45				
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45		
	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	-45	
	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90
	45																							

Figure 16: Test case 2 (with proportions constraint): the upper part of the symmetric stacking sequences.

One can see the correlation between weight and buckling loads. First, the buckling load is more sensitive to the thicknesses that stacking sequences. With the first 50 evaluations, the weight decreases due to the decrease of the thicknesses and the buckling load follows the weight. When the value weight is stabilized after evaluation 50, the buckling load is also stabilized. The change in the buckling load due to the change of the stacking sequences is less than its change due to change in the thicknesses.
Second, if the buckling load is less than $c = 500$ this means that there is no couple (S, D) which gives an admissible buckling load with the current thickness. Therefore, the weight is increased in order to compensate the violation of the constraints. Note also that, each point of these graphs correspond to a panel which completely satisfies the blending and manufacturing constraints. The two cases give slightly the same optimal weight. However, the buckling load with the test case with the proportions constraint has a higher value because imposing 50% of 0^o increases the stiffness of the panel with a compression along the x axis.
The Von Mises stress distribution over the panel at the first evaluation, evaluation 100 and the last evaluation of the optimization are shown in Figures 17-19. The stress in the panel increases when the weight of the panel is reduced between evaluations 1 and 100. After the evaluation 100, the weight remains about its optimal value and the stacking sequences redistribute the stress over the panel in order to comply with the buckling load.
This paper does not provide a theoretical result concerning the sufficient conditions for the convergence. This will be the subject of a future work. Here, the stopping criterion is the total number of evaluations. In these examples, it is

found that 750 evaluations where required to solve the optimization problem. The jumps in $\mathscr{W}$ and $\mathscr{B}$ curves are due to the fact that it is a discrete optimization. A jump corresponds to an added or a removed ply from the panel. Therefore, these jumps cannot vanish with iterations like with a continuous optimization problem.
This cost is very competitive with the genetic algorithms approach where 400 evaluations per iteration where required in [27]. This shows the efficiency of the proposed method.

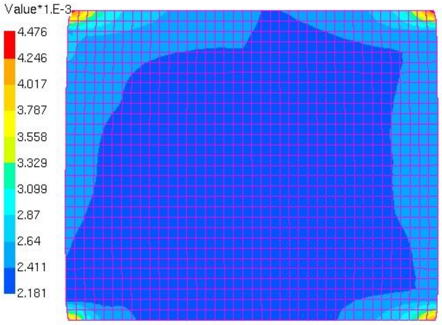


Figure 17: Test case 2 (with proportions constraint): the Von Mises stress (Pa) distribution over the panel at the first evaluation.

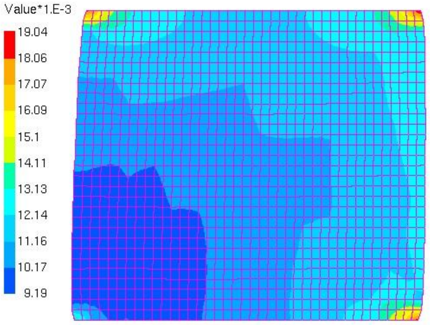


Figure 18: Test case 2 (with proportions constraint): the Von Mises stress (Pa) distribution over the panel at the evaluation 100.

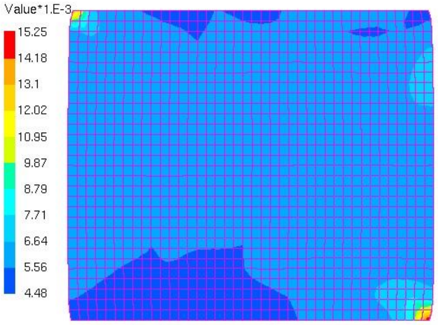


Figure 19: Test case 2 (with proportions constraint): the Von Mises stress (Pa) distribution over the panel at the last evaluation.

9. Conclusion

A new approach for the design of a composite structure was presented in this paper. Its novelty is that it seeks to minimize the weight of the structure such that a reserve factor (the buckling load) is higher than a threshold and the structure is manufacturable. The design variables are the thicknesses of regions of structure, their corresponding stacking sequences, and the ply drop-offs. The

optimization problem is formulated as a bilevel one where the thicknesses are sought together with the stacking sequences and the ply drop-offs.
A new optimization algorithm is proposed to solve this new approach. The novelty of this algorithm is that the optimization problem is solved witout splitting it into two steps as in common practice. The first advantage for the proposed optimization algorithm over the two steps one is the following. The thicknesses are updated according to the buckling load given by the current stacking sequences. The second advantage of the proposed optimization algorithm is the following. Bu using the backtracking method, at each iteration the structure completely satisfy the manufacturing rules. This new approach has been studied with a test case which showed its efficiency in terms of computational time and satisfaction of the manufacturability constraints.

References

[1] Boyang Liu, R. T. Haftka, M. A. Akgün and A. Todoroki, *“Permutation genetic Algorithm for Stacking Sequence Design of Composite Laminates”*, Computer methods in applied mechanics and engineering, vol. 186, pages 357-372 (2000).

[2] W. Liu, L. Krog, *“ A Method for Composite Ply Layout Design and Stacking Sequence Optimization”*, 7th ASMO UK / ISSMO conference on Engineering Design Optimization, Bath, UK July 7 - 8 (2008).

[3] A. Carpentier, *“Optimization multi-niveaux de panneaux composites”*, Ph.D. thesis (2008).

[4] A. Todoroki, and R.T. Haftka, *”Stacking sequence optimization by a genetic algorithm with a new recessive gene like repair strategy”*, Composites, Part B, 29(3), pp. pp. 277-285 (1998).

[5] D. Liu, D.C. Barton, V.V. Toropov, O.M. Querin, *“Optimum Stacking Sequence Design of a Blended Laminated Wing Structure using Permutation Genetic Algorithms (permGA)”*, B.H.V. Topping, J.M. Adam, F.J. Pallars, R. Bru, M.L. Romero, (Editors), ”Proceedings of the Tenth International Conference on Computational Structures Technology”, Civil-Comp Press, Stirlingshire, UK, Paper 125, 2010. doi:10.4203/ccp.93.125

[6] V. V. Toropov, R. Jones, T. Willment and M. Funnell, *“Weight and Manufacturability Optimization of Composite Aircraft Components Based on a Genetic Algorithm”*, 6th World Congresses of Structural and Multidisciplinary Optimization, Rio de Janeiro, 30 May - 03 June 2005, Brazil.

[7] G. Soremekun, Z. Gürdal, C. Kassapoglou and D. Toni *“Stacking Sequence Blending of Multiple Composite Laminates Using Genetic Algorithms”*, Composite Structures, vol. 56, pages 53-62 (2002).

[8] M. Bruyneel, *“A general and effective approach for the optimal design of fiber reinforced composite structures ”*, Composites Science and Technology, vol. 66, pages 1303-1314 (2006).

[9] D. B. Adams, L. T. Watson, O. Seresta and Z. Gürdal, *“Global/Local Iteration for Blended Composite Laminate Panel Structure Optimization Subproblems”*, Mechanics of Advanced Materials and Structures, vol. 14, pages 139 - 150 (2007).

[10] D. B. Adams, L. T. Watson and Z. Gürdal, *“Optimization and Blending of Composite Laminates Using Genetic Algorithms with Migration”*, Mechanics of Advanced Materials and Structures, vol. 10, pages 183 - 203 (2003).

[11] D. B. Adams , L. T. Watson, Z. Gürdal and C. M. Anderson-Cook, *“Genetic algorithm optimization and blending of composite laminates by locally reducing laminate thickness’*, Advances in Engineering Software, vol. 35, pages 35-43 (2004).

[12] S. T. IJsselmuiden, M. M. Abdalla, O. Seresta, Z. Gürdal, *“Multi-step blended stacking sequence design of panel assemblies with buckling constraints”*, Composites: Part B, vol. 40, pages 53-62 (2009).

[13] J. van Campen, O. Seresta, M. M. Abdalla and Z. Gürdal, *“General Blending Definitions for Stacking Sequence Design of Composite Laminate Structure”*, 49th AIAA/ASME/ASCE/AHS/ASC strutures, structural dynamics and Material conference, Schaumburg, April (2008).

[14] D. Liu, V.V. Toropov, O.M. Querin and D.C. Barton, *”Bi-level Optimization of blended Composite Panels”*, journal of aircraft vol.48(1), pages 107-118 (2011).

[15] D. Liu, V.V. Toropov, O.M. Querin and D.C. Barton, *”Stacking sequence optimization of composite panels for blending characteristics using lamination parameters”*, Paper 1687, 8th World Congress on Structural and Multidisciplinary Optimization, June 1 - 5, Lisbon, Portugal (2009).

[16] R. T. Haftka and J. L. Walsh, *”Stacking sequence optimization for buckling of laminated plates by integer programming”*, AIAA journal, Vol. 30 No. 3 (1992).

[17] G. J. Kennedy and J. R. R. A. Martins,*“A Laminate Parametrization Technique for Discrete Ply Angle Problems with Manufacturing Constraints”*, Structural and Multidisciplinary Optimization (in press).

[18] G. J. Kennedy and J. R. R. A. Martins, *“A Comparison of Metallic and Composite Aircraft Wings Using Aerostructural Design Optimization”*, in 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Indianapolis, IN, AIAA-2012-5475 (2012).

[19] M. Bruyneel, C. Beghin, G. Craveur, S. Grihon and M. Sosonkina "Stacking sequence optimization for constant stiffness laminates based on a continuous optimization approach", in Structural and Multidisciplinary Optimization, Volume 46, Issue 6, pp 783-794 (2012).

[20] S. Zein, B. Colson and S. Grihon *"A Primal-Dual Backtracking Optimization Method for Blended Composite Structures"*, Structural and Multidisciplinary Optimization, Volume 45, Number 5 (2012).

[21] M. Bruyneel, C. Beghin, G. Craveur and S Zein , *"Comparison of some solution methods for optimal stacking sequence of composites"*, in Proceedings of the European Conference on Spacecraft Structures, Materials & Environmental Testing, Noordwijk, Netherlands (2012).

[22] B. Liu, R.T. Haftka and M.A. Akgün, *"Two-level composite wing structural optimization using response surfaces "*, structural and Multidisciplinary Optimization Vol. 20, Number 2 (2000).

[23] W. J. ROUX, N. Stander and R. T. Haftka, *"Response surface approximations for structural optimization"*, Int. J. Numer. Meth. Engng. 42, 517-534 (1998).

[24] S. Russell and P. Norvig, *"Artificial Intelligence: A Modern Approach"*, Prentice Hall (2009).

[25] L. A. Wolsey, *"Integer Programming "*, chapter 12, Wiley-Interscience (1998).

[26] Donald L. Kreher and D. R. Stinson, *"Combinatorial Algorithms: Generation, Enumeration, and Search"*, chapter 4, CRC Press (1998).

[27] O. Seresta, Z. Gürdal, L. T. Watson, and D. B. Adams *"Optimal design of composite wing structures with blended laminates"*, Composites Part B: Engineering, Volume 38, Issue 4, pages 469-480 (2007).