A PRIMAL-DUAL BACKTRACKING METHOD FOR THE OPTIMIZATON OF BLENDED COMPOSITE STRUCTURES

S. Zein[1*], M. Bruyneel[1]

[1]*LMS Samtech, Liège Science Park, 8 rue des chasseurs ardennais, 4031 Angleur Belgium.*
**Samih.zein@lmsintl.com*

Keywords: Combinatorial Optimization; Backtracking; Composite Structures.

Abstract

A combinatorial optimization method is proposed for finding the optimal stacking sequence and the ply drop-offs scheme of a blended composite structure. This method assumes that the thicknesses of the regions in the structure are fixed in advance. It is able to handle efficiently design and manufacturing rules which are of combinatorial type. The optimization problem is formulated as a constrained binary programming problem and it is solved by applying both a primal and dual backtracking procedures with a local search method. Some numerical experiments are carried out to show the efficiency of the optimization method with respect to both computational time and quality criteria**.**

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 cover all the surface of the panel 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.

In this paper, we suppose that the spatial distribution of the thickness over the panel is fixed and we are interested in finding the ply drop-offs scheme and the fiber orientations that maximize the buckling load of the panel. It is further assumed that the fiber orientation in each ply can take one of the following values: -45°, 0°, 45° and 90°. The angle sequences of each region of the panel have to satisfy some design and manufacturing rules based on mechanical considerations. The angle sequences must contain a fixed number of plies of each orientation; two consecutive angles cannot have a difference of 90° and there can be at most four consecutive identical orientations. These constraints are called the design rules. The continuity of the plies in all regions of the panel is referred to as the manufacturing rules. The nature of such rules makes the optimization problem a combinatorial one.

Many optimization methods based on genetic algorithms have been developed to address this specific problem. They differ by the techniques used to satisfy the design/manufacturing rules. In [4,6,8,10,12] the manufacturing constraints are addressed using a penalty approach. In [2,9,11] 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 [1,3,5,7] the continuity of the plies (the blending) is imposed by a guide-based approach. The contribution in the present work can be summarized in three points.

- Defining a parameterization vector D of the ply drop-offs between the regions of the panel. This parameterization insures that the panel is always blended thus it avoids the use of a penalty. The sequences of all the regions can be deduced from two vectors: the angle sequence S of the thickest region and the ply drop-offs D. They are the only optimization variables.

- A mathematical formulation of the constraints (the design rules) is proposed based on a primal-dual approach. This approach gives the relationship between the angle sequences of the regions and the ply drop-offs in order to have an admissible panel at each iteration.

- An optimization algorithm based on the preceding two points is proposed. It uses a backtracking procedure with a local search method to perform the optimization. This standard choice in combinatorial optimization shows the efficiency of the proposed algorithm for the problem of our interest.

2 Parametrization of the Ply Drop-Offs and the Design Rules

Consider a panel composed of six regions, each one having its own thickness (see Figure 1). Let A and B be two adjacent regions where the thickness of B is smaller than or equal to the thickness of A. If the panel is blended, the plies composing region B have to be a subset of the ones of region A: some plies of region A are prolonged into region B while the others are dropped. The drop-off rules are not fixed in advance but they are parametrized by the permutation vector D. This permutation vector D allows defining any ply drop-offs.

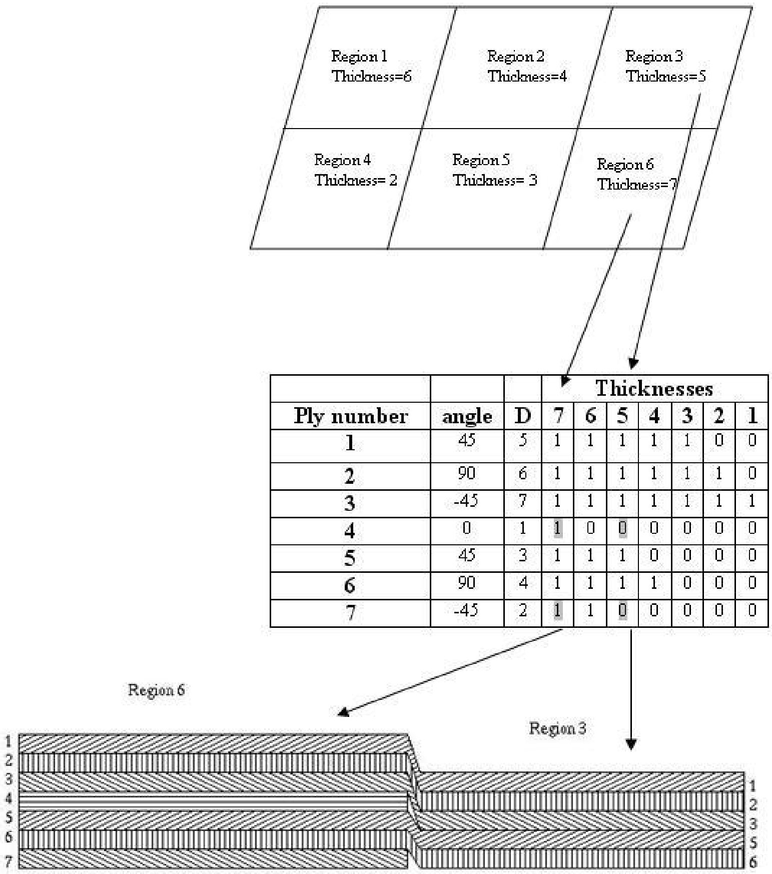
Let N be the number of plies in the thickest region. Using this vector D, a N×N lower triangular matrix can be constructed and its rows are permuted according to D. This matrix gives the drop-off rules following the blending principle. It gives the set of plies composing each possible thickness. Each column of this matrix represents a possible thickness and gives the ply sequence composing it. An element (i, j) of this table indicates whether ply i belongs to the set of plies of thickness j or not. 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 is composed of the plies (1, 2, 3, 4, 5, 6, 7) and region 3 of thickness 5 is made of the plies (1, 2, 3, 5, 6). The plies that 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. Using this table, one can deduce the ply sequences in all the regions of the panel and the blending principle is satisfied thanks to the fact that the drop-off table contains a permuted lower triangular matrix. 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.

The inward and the outward drop-off are special cases for which D = (1, 2, .., 7) and D = (7, 6, .., 1). 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 have to 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 orientations equal to 90° meaning that (0°, 90°) or (−45°, 45°) 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 interlaminar shear strength between groups of same orientation.
- A fixed number of plies of each orientation is defined in each region. This constraint is typically found in a global/local optimization framework. Note that this rule includes the case of a balanced sequence.



			Thicknesses						
Ply number	**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. Blending principle and drop-off rules defined by the permutation vector D.

3 The Primal and the Dual Problems

First, we define the following primal problem: for a given ply drop-offs scheme D, find all sequences S that are admissible with respect to D. Let A(D) be the set of all admissible sequences with respect to the ply drop-offs scheme D.

We define the following dual problem: for a given sequence S, find all the ply drop-offs schemes D for which S is admissible. In this case we use A'(S) to denote the dual set of all the ply drop-offs for which S is admissible.

The enumeration of the elements of these sets is based on backtracking algorithms which will be presented in Section 5. We also define the distance $d(S_1, S_2)$ between two sequences as the number of plies of S1 which have a different orientation in S_2, and the distance $d(D_1,D_2)$ between two ply drop-offs as the number of elements of D_1 which have a different value in D2. Such distances over discrete objects (like S and D) are known as the Levenshtein distances. We finally define the neighborhood of a sequence S1 A(D) and the one of a ply drop-off D_1 in A'(S) as:

$$V_{d0}^{D}(S_0) = \{S_1 | d(S_0, S_1) \leq d_0 \text{ and } S_1 \in A(D)\}$$
$$V_{d0}^{S}(D_0) = \{D_1 | d(D_0, D_1) \leq d_0 \text{ and } D_1 \in A'(S)\} \quad (1)$$

for some predefined d_0. The first one is the set of admissible sequences with respect to D that differ from S_1 by d_0 plies. The second one is the set of ply drop-offs for which S is admissible and differs from D_1 by d_0 elements.

4 The Primal and the Dual Problems

Finding admissible sequences 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 sequences which are admissible for a given ply drop-off is the so-called brute-force enumeration. It consists in enumerating all the sequence candidates 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 16 plies there are 4^{16}= 4294967296 candidates to be checked and for N=32 plies there are 4^32 ~1.844×10^{19} possibilities! A more sophisticated technique has to be used in order to decrease the number of candidates to be checked.
Enumerating all possible sequences consists in building an enumeration tree like in Figure 2.
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 the number of plies +1 level. A stacking sequence 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 the number of plies 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 the entire sub tree derived from the current node is eliminated from the enumeration tree. This pruning technique reduces considerably the size of the tree and makes the enumeration efficient.
For example in Figure, all the sub-sequences starting with (-45, 45), (0, 90), (45, -45) and (90, 0) are eliminated from the tree because they violate the 90° gap rule. The leaves of the tree are only the admissible sequences. The optimization algorithm is based on the backtracking procedure with a local search one. For more information, see [3].

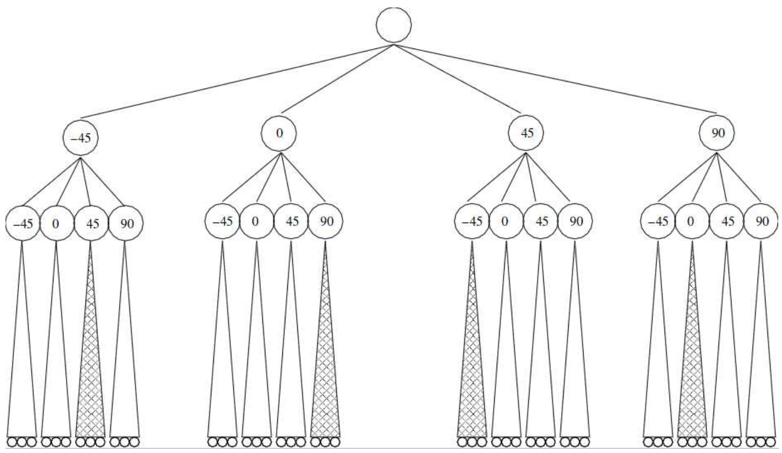


Figure 2. The enumeration tree.

5 The numerical algorithm

We define the following optimization problem:

$$\max_{S,D} F(S,D)$$
$$s.t.\ S \in A(D),$$

where S is the angle sequence of the thickest region and D is the ply drop-offs. F is some objective 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 3). First, D_k is fixed and S_k is updated. Following the definition of neighborhoods in section 3 and for some integer d_0, a set of admissible random sequences is generated in $V_{d0}^{Dk}(S_k)$, the neighborhood of S_k that comply with D_k. F is computed for all the sequences of the set. Then, S_{k+1} is assigned to the element of the set that has the highest value of F since F must be maximized. Next, a set of admissible random drop-offs schemes is generated in $V_{d0}^{Sk+1}(D_k)$, the neighborhood of D_k for which S_{k+1} is admissible. F is computed for all the sequences of the set. Then, D_{k+1} is set to the element of this set that has the highest value of F. The operation of generating random elements in the neighborhood of the current iteration is called local search and it relies on the backtracking procedure described in the previous section.

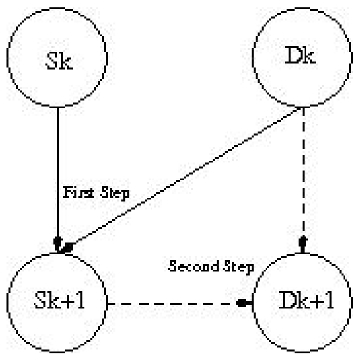


Figure 3. Illustration of the two steps to update (S,D) at each iteration.

The function F in (1) can be the buckling load of the composite structure or any other function, such as the one presented in the next section.

6 Definition of the degree of difference between two panels

Consider the stacking sequences S1 and S2 with a different number of plies. The Levenshtein distance between S1 and S2 gives the degree of difference between these two stacking sequences. It is the number of operations needed to transform S1 into S2. The operations are insert, delete and replace. For example, let S1=(-45,0,45,90) and S2=(0,0,45,0,90). The distance between S1 and S2 is 2 because the -45 is replaced by 0 and a 0 is inserted before the 90. This distance gives an idea on how much S1 is different from S2. If S1=S2, then the distance between them is zero. This notion of distance is generalized to the distance between panels. If two panels are composed of the same number of regions N and their stacking sequences are defined by (S1,D1) and (S2,D2), the distance between these two panels is the sum over the regions, of the distances between the two stacking sequences of the two panels. Note that if the two panels have the same number of plies per region, the distance between them is zero.

7 Numerical example

The numerical example consists in considering a "panel1" with defined thicknesses and stacking sequences, and a "panel2" with known thicknesses only. The goal of the optimization is to find the stacking sequences of panel2 which satisfy the design rules and which are the most similar to the ones of panel1. Thus, panel2 will have a mechanical behavior quite similar to the one of panel1, but with different thicknesses. The similarity between sequences is defined as the Levenshtein distance described in section 5. The goal is then to minimize the number of operations explained in section 5 that are necessary to go from the stacking sequences of panel1 to the ones of panel2.

Consider the following two panels composed of 8x6 regions. Thicknesses of the regions in number of plies in each panel are the following:

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 4. Panel with two sets of thicknesses

The percentages of 20% of -45°, 50% of 0°, 20% of 45° and 10% of 90° are imposed to the number of plies in each region. The staking sequences of the first panel are illustrated in Figure 5. The columns in grey give the optimal stacking sequences for the number of plies corresponding to the different regions of panel1.

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	001																				
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 5. The stacking sequences of the first panel.

The optimization problem consists in finding the stacking sequences that comply with the thicknesses of the second panel and that minimize the "distance" between the two panels. The proposed algorithm is used with the distance function as an objective function. The evolution of the distance function with respect to the number of iterations is shown in Figure 6 and the optimal (S,D) are shown in Figure 7. In Figure 6, it is seen that starting with a distance of about 500, an optimum solution is obtained with around 150 operations needed between both panels. One can see that the proposed algorithm is able to solve efficiently this optimization problem by generating stacking sequences that satisfy the constraints at each iteration.

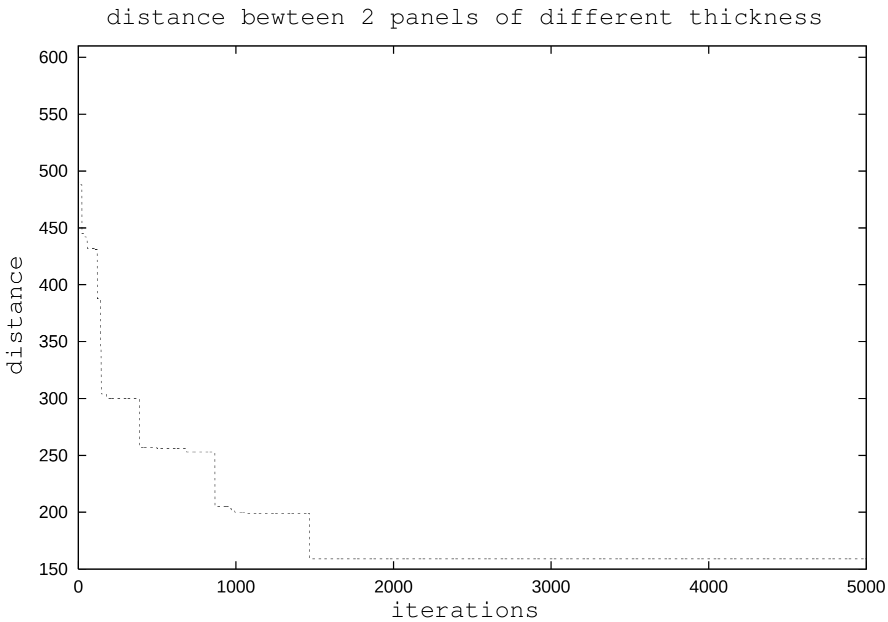


Figure 5. The Levenshtein distance with respect to the number of iterations.

8 Conclusions

In this paper, a combinatorial optimization method was proposed to find the optimal stacking sequence and the ply drop-offs scheme of a blended composite structure. Based on prescribed thicknesses in each region of the panel, the method can handle efficiently design and manufacturing rules which are of combinatorial type. The optimization problem was formulated as a constrained binary programming problem and it was solved by applying both a primal and dual backtracking procedures with a local search method. The efficiency of the method was demonstrated on an application.

9 Acknowledgements

Part of the work was done during the VIRTUALCOMP project funded by the Walloon Region of Belgium, under the supervision of Skywin (Aerospace Cluster of Wallonia).

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 6. The optimal stacking sequences of the second panel.

References

[1] 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).

[2] 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).

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

[4] L. A. Wolsey, “Integer Programming”, Wiley-Interscience (1998).

[5] D. L. Kreher and D. R. Stinson, “Combinatorial Algorithms: Generation, Enumeration, and Search”, CRC Press (1998).

[6] B. Liu, R.T. Haftka and M.A. Akgun, “Two-level composite wing structural optimization using response surfaces ”, structural and Multidisciplinary Optimization Vol. 20, Number 2 (2000).

[7] B. 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).

[8] Y. Radovcic and A. Remouchamps “BOSS quattro: an open system for parametric design”, Struct. Multidiscipl. Optim. vol 23, pages 140-152 (2002).

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

[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] 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).

[12] 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 structures, structural dynamics and Material conference, Schaumburg, April (2008).