Mixed Integer Linear Programming formulations of the stacking sequence and blending optimisation of composite structures

G. Ntourmas[a,b,*], F. Glock[b], F. Daoud[b], G. Schuhmacher[b], D. Chronopoulos[a], E. Özcan[c]

[a] *Institute for Aerospace Technology & The Composites Research Group, The University of Nottingham, NG7 2RD, UK*

[b] *Stress Methods and Optimisation, Airbus Defence and Space GmbH, 85077 Manching, Germany*

[c] *Computational Optimisation and Learning Lab, The University of Nottingham, NG7 2RD, UK*

Abstract

This manuscript proposes two novel formulations for the manufacturable stacking sequence retrieval of laminated composite structures. Detailed sizing of composite structures is commonly tackled by a two-stage optimisation approach, the first stage being a weight minimisation resulting in a continuous thickness and stiffness distribution of the structure. The second stage, which is the focus of the current paper, aims at retrieving stacking sequences that match the optimised target thickness and stiffness characteristics of the first stage, while also fulfilling a set of predefined design and manufacturing rules, with blending being one of the most crucial ones. The problem of retrieving layered designs that meet all of the prescribed composite rules is formulated as two Mixed Integer Linear Programming instances which

[*] Corresponding author

Email address: georgios.ntourmas@nottingham.ac.uk (G. Ntourmas)

mainly differ in how the blending is treated. Using either of the two formulations, mathematical programming algorithms can be employed to solve the problem to global optimality. Results show that both formulations can produce manufacturable solutions that fulfil the imposed composite rules. High-quality solutions can be obtained consistently, while benefiting from increased design freedom concerning the blending formulation.

Keywords: Integer Programming, Blending, Stacking Sequence Optimisation

1. Introduction

Composite materials, such as fibre-reinforced plastics, find application in many high-end products such as aeronautical or automotive structures, wind turbines, sports equipment and more because they offer reduced weight and enhanced mechanical characteristics. Aside from this, composites are generally anisotropic materials, providing the engineer with increased design freedom when compared to the classic metallic materials. However, the problem that arises is that with increased design freedom, the complexity of sizing the structure in detail also increases.

Most high performance aircraft structures are produced by laying unidirectional tapes of polymer pre-impregnated fibres one next to each other to form a lamina or ply with a specific fibre orientation. More plies are added on top, in such a way that their orientation and number forms a laminate which satisfies the stiffness and thickness requirements of the part. Large-scale structures are discretised in so-called patches across their span to address the local requirements and therefore the stacking sequence varies between the patches.

During the optimisation process, a large set of constraints must be taken into account. First of all, design rules regulating the stacking characteristics of each patch in the structure should be followed. Secondly, to ensure structural integrity and manufacturability, the stacking sequence of neighbouring patches needs to be integrated into a design that follows specific rules, referred to as manufacturing rules, in which blending or continuity of the plies is of significant importance. Finally, the structure needs to fulfil a large set of physical constraints, related to strength, buckling, aeroelasticity, damage tolerance and other requirements.

The problem of composite stacking sequence optimisation has been studied thoroughly by many researchers over the last decades. A review of different approaches to stacking sequence optimisation can be found in Ghiasi et al. [15, 14]. Two particularities of the problem dominate the nature of the proposed optimisation approaches. Firstly, the problem contains mostly integer design variables but both discrete and continuous constraints. Secondly, many of the continuous constraints require computationally expensive Finite Element (FE) model evaluations. Therefore, although metaheuristics have been widely employed to perform the optimisation [26, 46, 34, 9, 4, 42, 22, 3], their usage becomes prohibitive for large-scale problems due to the increased number of optimisation cycles required, combined with the computational expense of the FE evaluations. Furthermore, gradient-based optimisation methods [45, 31, 44, 6, 13, 25] are very well suited for the continuous part of the problem, combined with the solution cost of the FE model, but fail to efficiently handle the discrete part of the problem.

In an attempt to incorporate the benefits and eliminate the drawbacks of both gradient and stochastic algorithms, a two-stage optimisation process has been employed by several researchers, previously. During the first stage, a weight minimisation is performed [18, 36] using either lamination [19, 32] or polar parameters [35, 37] to describe the structural stiffness. The result of this stage is an optimal, continuous distribution of thickness and stiffness characteristics. The continuous thickness is usually rounded up to the nearest integer number of layers in order to achieve a manufacturable design. This discretisation is performed since the thickness of the pre-impregnated tape that will be used in manufacturing, is decided *a priori* and is treated as a constant parameter during the sizing optimisation. This discrete thickness and the aforementioned stiffness distribution, can be used as targets by the second stage of the optimisation process which is usually a stochastic algorithm [28, 20, 7, 23] that can handle design and manufacturing rules more efficiently.

The principal novelty of this work is the formulation of the composite manufacturing rules as two different Mixed Integer Linear Programming (MILP) models. Although potential structural engineering applications of operational research have been identified [27], a complete MILP formulation of the stacking sequence optimisation has not yet been presented to the best of the authors' knowledge with the exception of some composite design rules which have been previously formulated in such a way [2]. Additionally, Kang & Blom [24] have worked towards that direction, having however treated the blending rules in the context of compliance with pre-computed laminates, which is not the case for the present study. Other studies focusing on tree structures, branch and bound algorithms and beam search algorithms [40, 39, 50, 29, 30, 10, 11], are algorithmic approaches built around the specific nature of the stacking sequence optimisation problem in question. In this work, two complete formulations accounting for all composite design and manufacturing rules which can be used in any optimisation solver able to handle a MILP problem formulation are presented.

The rest of this paper is structured as follows: In Section 2 composite guidelines relevant to the stacking sequence retrieval for aeronautical structures are presented. The two MILP formulations of the blended stacking sequence optimisation are formulated in Section 3. Results from the two formulations are demonstrated in Section 4 and the findings of this work are summarised in Section 5.

2. Composite rules

Besides physical constraints such as buckling, strength, maximum allowable displacements etc., that naturally govern the optimisation problem, there are also several rules, which are commonly used in the aerospace industry when designing composite structures. These rules serve as guidelines for manufacturing laminates, which are less prone to high stress concentration and unwanted mechanical coupling effects. Most of these rules have also been presented in previous works [21, 5]. Below, the composite rules implemented and used in this work are summarised.

The composite rules are grouped into two categories, i.e. design and manufacturing rules. Design rules regulate the stacking sequence of a single uniform laminate.

1. *Symmetry.* Symmetric laminates about the mid-plane are commonly used to avoid bending-extension coupling.

2. *Balance.* Balanced laminates consist of equal $+\theta$ and $-\theta$ orientations ($\theta \neq 0^o, 90^o$), in order to eliminate shear-extension coupling.

3. *Minimum percentage.* A minimum percentage of all fibre orientations used in a laminate might be enforced to minimise matrix degradation and favour a fibre-dominated failure mode instead.

4. *Damage tolerance.* External plies should never be in the direction of the main load path. Commonly, a 45^o or -45^o layer is placed in the outermost part of the laminate.

5. *Contiguity.* The maximum grouping of equally oriented layers is limited in order to minimise interlaminar stresses and ensure a homogeneous stress distribution.

6. *Grouping.* Grouping $+\theta$ and $-\theta$ layers might be applied to reduce bending-twist coupling.

7. *Disorientation.* The difference in fibre orientation between two adjacent layers might be limited to 45^o to minimise inter-laminar shear effects.

It should be noted that although grouping and disorientation design rules may both be advisable for a structure, their requirements are contradictory and hence cannot be applied simultaneously.

The manufacturing rules consider the transitioning between laminates placed in neighbouring patches.

1. *Continuity/blending.* Continuity, also commonly referred to as blending, ensures manufacturability and structural integrity of the laminated composite. Various definitions of blending are available in the literature. Adams et al. [1] used two simplified blending definitions, i.e. outward (Fig. 1a) and inward (Fig. 1b) blending, in which only outermost or innermost plies, respectively, are allowed to be dropped. A ply drop indicates that a ply stops and does not continue on to the next patch of the structure. Campen et al. [8] introduced generalised (Fig. 1c and Fig. 1d) and relaxed generalised (Fig. 1e) blending. These two definitions allow for significantly greater design freedom since they do not limit the ply drop position to an outer or inner ply. Generalised blending requires all plies in a thinner patch to continue in the adjacent thicker panels, while in relaxed generalised blending two neighbouring panels are considered to be blended if no dropped edges are in physical contact. In the current work, generalised blending is used instead of the relaxed generalised blending, as the latter leads to high stress concentrations.

2. *Maximum dropping.* The maximum number of plies dropped in transitions between neighbouring laminates is limited to ensure smooth load distribution throughout the structure.

3. *External covering ply.* At least one of the outer plies in a laminate is not allowed to be dropped in order to ensure structural integrity.

4. *Internal covering ply.* Not more than a certain number of consecutive plies are allowed to be dropped simultaneously so that potential

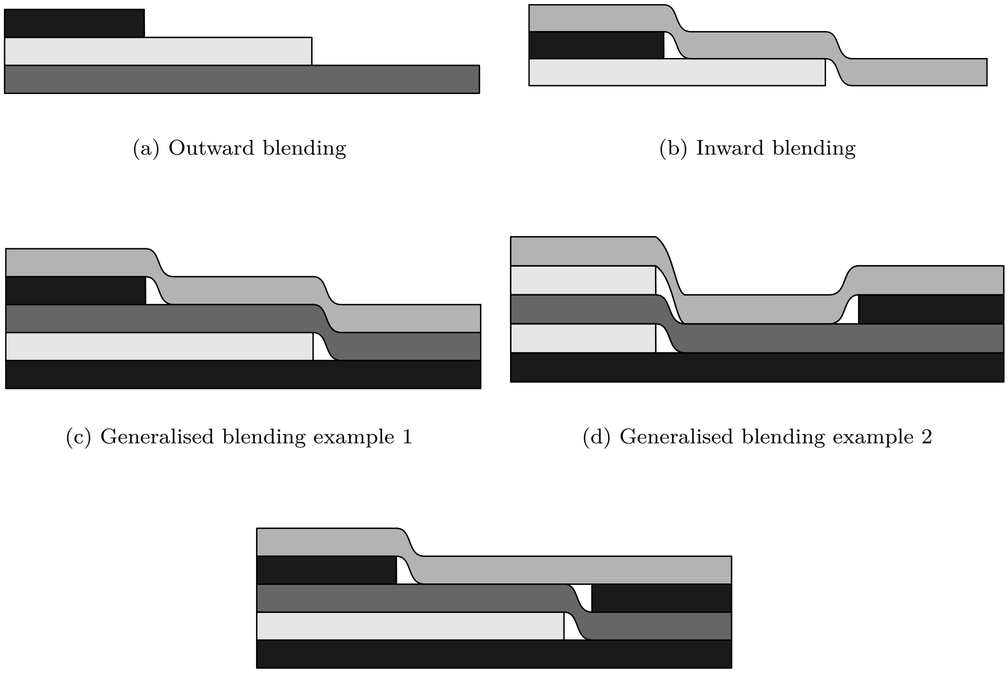
delamination initiation zones are avoided.

Other manufacturing rules concerning the exact transitions within the taper area are not considered, because this level of detail is not in the scope of the current work. It should be mentioned that the taper area refers to the zone formed by the plies of two neighbouring patches merging into each other.

The full potential of the generalised blending definition is exploited in the current work (Fig. 1d), in a sense that the set of plies used in the thinnest patch must not necessarily extend to the rest of the patches in the structure. This is not the case for other implementations of generalised blending, such as the Stacking Sequence Tables (SSTs) [21], the Global Shared Layer Blending [23] or the modified Shepard's method [33], in which plies present in the thinnest patch must be continued across all other patches and not only the neighbouring ones (Fig. 1c). However, it should be noted that recent studies using the Ply Drop Sequence concept [49], multiple SSTs [51] and a general resolution scheme [38] have managed to eliminate some or all of the aforementioned drawbacks, offering increased design freedom.

3. Stacking sequence retrieval

In this study, a two-stage optimisation process for the detailed sizing of aerospace composite structures is investigated. First, a weight minimisation of the structure is performed using a gradient-based optimisation algorithm, the result of which is a continuous thickness and stiffness distribution. A given structure is manually discretised into patches, which represent areas that will be considered as individual laminates during the optimisation. This gradient-based optimisation is part of the multidisciplinary optimisation tool,



(a) Outward blending

(b) Inward blending

(c) Generalised blending example 1

(d) Generalised blending example 2

(e) Relaxed generalised blending

Figure 1: Blending examples between 3 patches.

referred to as *LAGRANGE* [41], developed in-house at Airbus. In the second stage of the process, which is the focus of this work, the stacking and blending optimisation problem is formulated as two different MILP models, which are solved using Gurobi [17]. The resultant stacking sequence, fulfils all design and manufacturing rules by matching the optimal stiffness characteristics calculated during the first optimisation stage. These stiffness characteristics are provided to the second stage of the optimisation in the form of lamination parameters.

3.1. Lamination parameters

Lamination parameters were first introduced by Tsai & Pagano [47]. They can be used to decouple the stacking-sequence-dependent part from the material-dependent part of a laminate's stiffness matrix. In the general case of an anisotropic laminate, 12 lamination parameters and 5 material parameters fully define the stiffness matrix. The extensional stiffness **A**, bending extension coupling stiffness **B** and bending stiffness **D** matrices are formulated as:

$$\begin{Bmatrix} A_{11} \\ A_{22} \\ A_{12} \\ A_{66} \\ A_{16} \\ A_{26} \end{Bmatrix} = h \begin{bmatrix} 1 & \xi_1^A & \xi_3^A & 0 & 0 \\ 1 & -\xi_1^A & \xi_3^A & 0 & 0 \\ 0 & 0 & -\xi_3^A & 1 & 0 \\ 0 & 0 & -\xi_3^A & 0 & 1 \\ 0 & \xi_2^A/2 & \xi_4^A & 0 & 0 \\ 0 & \xi_2^A/2 & -\xi_4^A & 0 & 0 \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \end{Bmatrix} \tag{1}$$

$$\begin{Bmatrix} B_{11} \\ B_{22} \\ B_{12} \\ B_{66} \\ B_{16} \\ B_{26} \end{Bmatrix} = \frac{h^2}{4} \begin{bmatrix} 0 & \xi_1^B & \xi_3^B & 0 & 0 \\ 0 & -\xi_1^B & \xi_3^B & 0 & 0 \\ 0 & 0 & -\xi_3^B & 0 & 0 \\ 0 & 0 & -\xi_3^B & 0 & 0 \\ 0 & \xi_2^B/2 & \xi_4^B & 0 & 0 \\ 0 & \xi_2^B/2 & -\xi_4^B & 0 & 0 \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \end{Bmatrix} \tag{2}$$

$$\begin{Bmatrix} D_{11} \\ D_{22} \\ D_{12} \\ D_{66} \\ D_{16} \\ D_{26} \end{Bmatrix} = \frac{h^3}{12} \begin{bmatrix} 1 & \xi_1^D & \xi_3^D & 0 & 0 \\ 1 & -\xi_1^D & \xi_3^D & 0 & 0 \\ 0 & 0 & -\xi_3^D & 1 & 0 \\ 0 & 0 & -\xi_3^D & 0 & 1 \\ 0 & \xi_2^D/2 & \xi_4^D & 0 & 0 \\ 0 & \xi_2^D/2 & -\xi_4^D & 0 & 0 \end{bmatrix} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \end{Bmatrix} \tag{3}$$

In the above equations, $\mathbf{U}$ denotes the material constants and h the thickness of the laminate. The lamination parameters are defined as:

$$\xi^A_{[1,2,3,4]} = \frac{1}{h}\int_{-\frac{z_i}{2}}^{\frac{z_i}{2}} [\cos(2\theta), \sin(2\theta), \cos(4\theta), \sin(4\theta)]\mathrm{d}z \tag{4}$$

$$\xi^B_{[1,2,3,4]} = \frac{4}{h^2}\int_{-\frac{z_i}{2}}^{\frac{z_i}{2}} [\cos(2\theta), \sin(2\theta), \cos(4\theta), \sin(4\theta)]z\mathrm{d}z \tag{5}$$

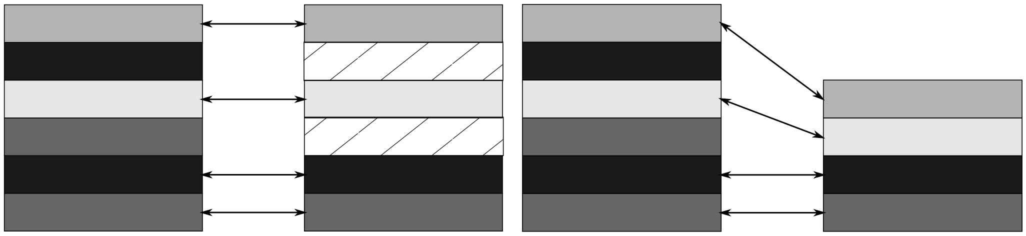
$$\xi^D_{[1,2,3,4]} = \frac{12}{h^3}\int_{-\frac{z_i}{2}}^{\frac{z_i}{2}} [\cos(2\theta), \sin(2\theta), \cos(4\theta), \sin(4\theta)]z^2\mathrm{d}z \tag{6}$$

where $-\frac{z_i}{2}$ and $\frac{z_i}{2}$ stand for the distance of the bottom and top surface of the i^{th} ply with respect to the midplane of the laminate.

3.2. Mixed Integer Linear Programming (MILP) formulation

The input for the stacking sequence calculation is a simple geometric definition of the patches, together with the optimal number of the total layers and lamination parameters for each patch. The MILP formulation of this problem is discussed in this section. Two distinct formulations of the blended stacking sequence optimisation, namely explicit and implicit, have been developed. As far as the design rules are concerned, their expression does not fundamentally differ between the two formulations. The distinctive difference between the two is how blending and ply drops are modelled. Nevertheless, both the explicit and implicit formulation make use of the same principles of generalised blending described in Section 2. Therefore, the resulting structure from each formulation is expected to be the same if the same input and constraints are used for each optimisation.

In brief, the explicit formulation considers all patches having the same number of layers, with that number being equal to the number of layers in the thickest patch of the structure. Since each patch has a different optimum number of layers, some of the patches will consist of plies that correspond to ply-voids as displayed in Fig. 2a. The implicit formulation demonstrated in Fig. 2b takes into account the fact that each patch has a different, known number of optimum layers, and defines the appropriate number of layers for each individual patch. The benefit of the explicit formulation is that the blending of the structure is rather straightforward, as continuity is ensured by placing plies of the same orientation one next to the other. The disadvantage of this formulation is that the locations of the ply drops are not known and therefore there are many possible 'positions' one specific ply could have within the laminate. Therefore, since the order at which plies will appear is not known, the calculation of the lamination parameters for such a stack is cumbersome. The advantage of the implicit formulation is that the calculation of lamination parameters is straightforward as the position of each ply is clearly defined. The downside of the method is the fact that formulating the manufacturing rules and most importantly the continuity of the structure is more challenging.



(a) Schematic representation of the explicit formulation.

(b) Schematic representation of the implicit formulation.

Figure 2: An illustrative example of the same blended stack for two neighbouring patches as represented by the explicit and implicit formulation. Each colour represents a different fibre orientation, ply-voids are illustrated using hatched plies and arrows indicate ply continuity.

3.2.1. Explicit formulation

Various design variables are introduced in order to formulate the blended stacking sequence problem. Each set of design variables, i.e. x, y, b, c, d, z, u, g, f contains many individual members. The number of members for each set depends on the usage of the design variable and different indexes are used to distinguish between the individual members of each set of design variables. Index $i \in \{1, 2, ..., I\}$ is used to denote the exact layer in a specific patch $j \in \{1, 2, ..., J\}$. It is important to mention that the range of i is constant for all patches j, even if most of the patches are expected to have less layers than I which is the number of layers for the thickest patch in the structure. Index $\theta \in \{1, 2, ..., \Theta\}$ denotes the different available fibre orientations. Finally, $k \in \{1, 2, 3, 4\}$ distinguishes between the lamination parameter which is of interest. A summary of the indexes employed to differentiate between the members of each design variable set is given in Table 1.

The objective is to minimise the deviation from the optimal values of the

Table 1: Summary of indexes used to denote the various design variables in the explicit formulation.

Name	Description
i	Layer
j	Patch
θ	Fibre orientation
k	Lamination parameter

lamination parameters as formulated below:

$$\min \sum_{j=1}^{J} w_k^{A,B,D} |\xi_{kj}^{A,B,D} - (\xi_{kj}^{A,B,D})_{\text{optimal}}| \quad \forall k \tag{7}$$

where $\xi_{kj}^{A,B,D} \in [-1, 1]$ are the lamination parameters calculated for a specific stack, whereas, $(\xi_{kj}^{A,B,D})_{\text{optimal}}$ are the optimal parameters delivered by the gradient-based optimisation. Weight coefficients $w_k^{A,B,D}$ may be used to emphasise specific parts of the objective function. Two new sets of design variables need to be introduced to assist with the linear formulation of the objective function. The first one is $g_{kj}^{A,B,D} \in [0, 1]$, defined as $g_{kj}^{A,B,D} = |\xi_{kj}^{A,B,D} - (\xi_{kj}^{A,B,D})_{\text{optimal}}|$. The objective function is not linear in its current form, because of the absolute value, therefore, a simple linearisation needs to be performed. The additional set of design variables $f_{kj}^{A,B,D} \in \{0, 1\}$, is also introduced to enable the linearisation described by the following set of equations

$$\begin{aligned}
&\xi_{kj}^{A,B,D} - (\xi_{kj}^{A,B,D})_{\text{optimal}} \leq f_{kj}^{A,B,D} \quad \forall j, k \\
&(\xi_{kj}^{A,B,D})_{\text{optimal}} - \xi_{kj}^{A,B,D} \leq 1 - f_{kj}^{A,B,D} \quad \forall j, k \\
&\xi_{kj}^{A,B,D} - (\xi_{kj}^{A,B,D})_{\text{optimal}} \leq g_{kj}^{A,B,D} \quad \forall j, k \\
&(\xi_{kj}^{A,B,D})_{\text{optimal}} - \xi_{kj}^{A,B,D} \leq g_{kj}^{A,B,D} \quad \forall j, k \\
&g_{kj}^{A,B,D} \leq \xi_{kj}^{A,B,D} - (\xi_{kj}^{A,B,D})_{\text{optimal}} + 2(1 - f_{kj}^{A,B,D}) \quad \forall j, k \\
&g_{kj}^{A,B,D} \leq (\xi_{kj}^{A,B,D})_{\text{optimal}} - \xi_{kj}^{A,B,D} + 2f_{kj}^{A,B,D} \quad \forall j, k
\end{aligned} \tag{8}$$

To make the formulation of the problem possible, the binary design variables $x_{ij\theta} \in \{0, 1\}$ defining the orientation of a specific layer within a certain patch are introduced. More specifically, every layer consists of Θ design variables, each one representing whether the corresponding fibre orientation is used or not. Each layer may only have one orientation, or none, in the case the ply is dropped. Therefore, the following feasibility constraints need to be defined.

$$\sum_{\theta=1}^{\Theta} x_{ij\theta} \leq 1 \quad \forall i, j \tag{9}$$

Additionally, the number of layers in a patch must be equal to the optimum discretised thickness (n^j) which originates from the continuous optimisation and remains fixed throughout the discrete optimisation.

$$\sum_{i=1}^{I} \sum_{\theta=1}^{\Theta} x_{ij\theta} = n^j \quad \forall j \tag{10}$$

In many cases, laminates are designed to be symmetric. Whenever symmetry is required, the following constraints need to be incorporated in the optimisation model.

$$x_{ij\theta} = x_{(I-i+1)j\theta} \quad \forall i\{1, 2, ..., I/2\}, j, \theta \tag{11}$$

Balanced laminates can be extracted from the optimisation by including the following constraints.

$$\sum_{i=1}^{I} x_{ij\theta_1} = \sum_{i=1}^{I} x_{ij\theta_2} \quad \forall i, j, \theta_1 = -\theta_2 \neq \{0, 90\} \tag{12}$$

The minimum percentage (p) design rule is implemented as:

$$\sum_{i=1}^{I} x_{ij\theta} \geq pn^j \quad \forall j, \theta \tag{13}$$

The damage tolerance rule is demonstrated below for the case that the outermost ply is equal to 45^o. This layer also has to be continuous over all patches according to the external covering ply manufacturing rule. The following constraints fulfill both requirements.

$$x_{1j\theta_1} = 1 \quad \forall j, \theta_1 = 45 \tag{14}$$

The grouping design rule for the case of symmetric laminates is formulated as:

$$x_{ij\theta_1} \leq x_{ij\theta_2} \quad \forall i \in \{1, 2, ..., I/2\}, j, \theta_1 = -\theta_2 \neq \{0, 90\} \tag{15}$$

For non symmetric laminates, the i range needs to be modified accordingly. The contiguity and disorientation constraints cannot be easily implemented in the explicit formulation and are hence only going to be presented for the implicit formulation in the next section.

In order to define the continuity manufacturing constraint between neighbouring patches, further design variables need to be introduced. A set of them is $y_{ij} \in \{0, 1\}$, defining the presence or absence of a specific ply within a patch. The continuity constraint is expressed as:

$$x_{ij_2\theta} = x_{ij_1\theta} y_{ij_2} \quad \forall i, j, \theta. \tag{16}$$

Because the constraint presented above is non-linear, it needs to be linearised in order to be handled by a MILP solver. The linearisation constraints for the product of two binary variables [12] are now introduced.

$$\begin{aligned} & x_{ij_2\theta} \leq x_{ij_1\theta} \quad \forall i, j, \theta \\ & x_{ij_2\theta} \leq y_{ij_2} \quad \forall i, j, \theta \\ & x_{ij_2\theta} \geq x_{ij_1\theta} + y_{ij_2} - 1 \quad \forall i, j, \theta \end{aligned} \tag{17}$$

In the above equation, patches j_1 and j_2 denote two neighbouring patches within the structure. An additional constraint which is not necessary, but has been observed to assist the convergence of the optimisation, is used to define the number of ply drops per patch based on the known number of layers for each patch.

$$\sum_{i=1}^{I} y_{ij} = n^j \quad \forall j \tag{18}$$

Before moving on to the definition of the rest of the constraints, it is worth formulating the lamination parameter values in a way that serves the specific approach to the problem. The formal definition of lamination parameters has already been presented in the previous section (equations 4-6).

$$\xi_{kj}^{A,B,D} = \sum_{i=1}^{I} \sum_{\theta}^{4} s_{k\theta} a_{ij}^{A,B,D} x_{ij\theta} \quad \forall j, k \tag{19}$$

Coefficients $s_{k\theta}$ take into consideration the influence of the trigonometric terms, depending on which θ design variable is used out of $x_{ij\theta}$. Additionally, coefficients $a_{ij}^{A,B,D}$ depend on whether A, B or D lamination parameters are considered and on the position of the ply within the laminate. Due to the fact that ply drops are included in the problem formulation as design variables, the distance of each ply from the mid-surface of the laminate is not known *a priori*. Another set of design variables $b_{ij} \in \{0, 1, 2, ..., M_1\}$ is used for that purpose, where the upper bound is $M_1 = i + 1$ The definition of b_{ij} is as follows:

$$b_{ij} = \sum_{m=1}^{i} y_{mj} \quad \forall i, j \tag{20}$$

In summary, the aforementioned coefficients $a_{ij}^{A,B,D}$ depend on the b_{ij} design variables and are defined as:

$$a_{ij}^{A} = \frac{1}{n^j} \quad \forall i, j \tag{21}$$

$$a_{ij}^{B} = -\frac{4}{(n^j)^2} b_{ij} + \frac{2n^j + 2}{(n^j)^2} \quad \forall i, j \tag{22}$$

$$a_{ij}^{D} = \frac{12}{(n^j)^3} (b_{ij})^2 - \frac{12n^j + 12}{(n^j)^3} b_{ij} + \frac{3(n^j)^3 + 6n^j + 4}{(n^j)^3} \quad \forall i, j \tag{23}$$

The term $(b_{ij})^2$ in equation 23 is not linear and therefore needs to be linearised. Due to the fact that the linearisation of a product of two bounded integer variables is cumbersome and involves piecewise linearisations, a trick is employed to calculate $c_{ij} \in \{0, 1, 2, ..., (M_1)^2\}$ as:

$$(b_{ij})^2 = c_{ij} = (\sum_{m=1}^{i} 2d_{mj}) - b_{ij} \quad \forall i, j \tag{24}$$

In the above equation, a new set of design variables $d_{ij} \in \{0, 1, 2, ..., M_1\}$ was introduced. These are defined as $d_{ij} = b_{ij}y_{ij}$ and once again need to be linearised. The linearisation of the product of two variables, one of which is binary and the other an integer with an upper bound of M_1 and a lower bound of 0, is defined by the following four inequalities [16].

$$\begin{aligned} & d_{ij} \leq M_1 y_{ij} \quad \forall i, j \\ & d_{ij} \leq b_{ij} \quad \forall i, j \\ & d_{ij} \geq b_{ij} - (1 - y_{ij})M_1 \quad \forall i, j \\ & d_{ij} \geq 0 \quad \forall i, j \end{aligned} \tag{25}$$

Finally, due to the fact that the product of $a_{ij}^{B,D} x_{ij\theta}$ appears in equation 19, two new sets of design variables $u_{ij\theta} \in \{0, 1, 2, ..., M_1\}$, $z_{ij\theta} \in \{0, 1, 2, ..., (M_1)^2\}$ need to be introduced and are defined as $u_{ij\theta} = b_{ij}x_{ij\theta}$ and $z_{ij\theta} = c_{ij}x_{ij\theta}$. The linearisation of these design variables follows the principles of equation 25. The lamination parameter values can eventually be expressed as:

$$(\xi_{kj}^{B}) = \sum_{i=1}^{I} \sum_{\theta=1}^{4} s_{k\theta}(-\frac{4}{(n^j)^2} u_{ij\theta} + \frac{2n^j + 2}{(n^j)^2} x_{ij\theta}) \quad \forall k, j \tag{26}$$

$$(\xi_{kj}^{D}) = \sum_{i=1}^{I} \sum_{\theta=1}^{4} s_{k\theta}(\frac{12}{(n^j)^3} z_{ij\theta} - \frac{12n^j + 12}{(n^j)^3} u_{ij\theta} + \frac{3(n^j)^3 + 6n^j + 4}{(n^j)^3} x_{ij\theta}) \quad \forall k, j \tag{27}$$

The internal covering ply manufacturing constraint is formulated using the b_{ij} design variables as:

$$b_{(i+K)j_1} - b_{(i+K)j_2} - (b_{ij_1} - b_{ij_1}) \leq K - 1 \tag{28}$$

where j_1 and j_2 denote two neighbouring patches and K the number of maximum consecutive ply drops. The maximum dropping manufacturing rule is explicitly satisfied due to the fixed number of layers provided as an input to the optimisation and therefore does not require further constraints.

3.2.2. Implicit formulation

Similarly to the explicit formulation, various design variables need to be introduced. Each set of design variables, i.e. x, r, t, g, f, contains many individual members. Once again, index $i \in \{1, 2, ..., I_j\}$ is used to denote the exact layer in a specific patch $j \in \{1, 2, ..., J\}$. The range of i is not constant for all patches j, but rather takes a maximum value of I_j according to the patch of interest. Index $h \in \{1, 2, ..., H\}$ accounts for all interfaces between neighbouring patches and index $p \in \{1, 2, ..., P_h\}$ takes into account the different blending combinations between neighbouring patches. A more detailed explanation of these two indexes will be presented in the following paragraphs. Finally, indices θ and k follow the definition of the explicit formulation and denote fibre orientations and lamination parameter which is of interest respectively.

The formulation of the objective function is exactly the same as the one presented in equation 7 and the same goes for the set of design variables $g^D_{kj} \in [0, 1]$ and $f^D_{kj} \in \{0, 1\}$. The definition of the $x_{ij\theta} \in \{0, 1\}$ set of design variables follows the same logic with the exception that in the implicit formulation, each patch is assigned with as many I_j layers as needed. All design rules (equations 11-15) are formulated similarly with the only thing changing being the number of maximum layers per patch I_j. An additional minor alternation applies for equation 9 in which the inequality sign needs to be replaced by the equality one, since the design variables $x_{ij\theta}$ in this approach cannot correspond to a ply drop.

Additionally, the two design rules which have not been formulated for the explicit approach can now be implemented. The contiguity constraint limits the number of consecutive layers having the same fibre orientation to a maximum of N. In a mathematical formulation this is expressed as:

$$x_{ij\theta} + x_{(i+1)j\theta} + \ldots + x_{(i+N)j\theta} \leq N \quad \forall i \in \{1, 2, ..., I_j - N\}, j, \theta \tag{29}$$

The disorientation constraint is expressed as:

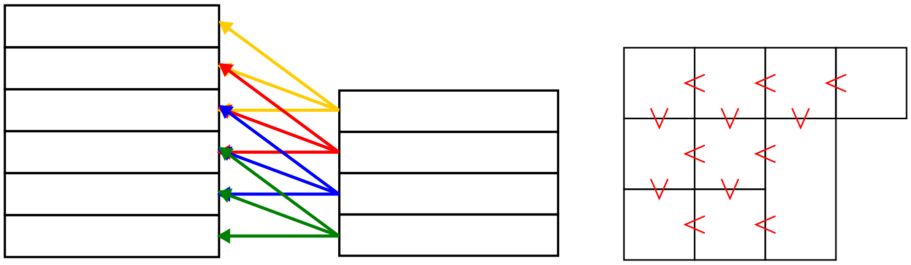
$$x_{ij\theta_1} + x_{(i+1)j\theta_2} \leq 1 \quad \forall i, j, |\theta_1 - \theta_2| \geq 45 \tag{30}$$

The calculation of lamination parameter values is now trivial. Equation 19 is once again used except this time, coefficients $a_{ij}^{B,D}$ can be easily pre-calculated due to the fact that ply drops have already been taken into account by using an exact number (I_j) of $x_{ij\theta}$ design variables for each patch.

The part of this second approach that becomes cumbersome is formulating the continuity between neighbouring patches. A new set of binary design variables $t_{ihp} \in \{0, 1\}$ is introduced and is used to regulate and keep track of whether a specific ply within a patch will be blended with one of the p possible neighbouring plies. The number of possibilities p per ply depends on the interface h examined and can easily be determined during the setup of the optimisation problem. A representation of the different plies that could blend between two patches is given in Fig. 3a. More specifically, for any of the interfaces h, the maximum number of possibilities per ply (P_h) can be calculated as:

$$P_h = |n_{j_1} - n_{j_2}| + 1 \tag{31}$$

In the above equation, patches j_1 and j_2 denote the two patches which constitute an interface h, with j_1 being the thickest of the two, in case there is a difference in thickness between them. The number of interfaces H is easily calculated prior to setting up the optimisation problem based on the arrangement of the patches. In Fig. 3b, an example of the interfaces between a specific patch geometry is given.



(a) Knowing the number of ply drops between two neighbouring patches, the different blending possibilities can be determined for each ply independently.

(b) The interfaces between patches are marked with red ticks for the given patch geometry.

Figure 3: Illustrative examples showing the usage of indexes p and h.

Finally, a new set of design variables $r_{ih\theta p} \in \{0, 1\}$ is defined as $r_{ih\theta p} = x_{(i+p-1)j_1\theta} t_{ihp}$. Because the definition of $r_{ih\theta p}$ is non-linear, a standard linearisation needs to be performed. This linearisation of the product of two binary variables is similar to the one presented in equation 17.

The continuity manufacturing rule is then defined as:

$$x_{ij_2\theta} = \sum_{p}^{P_h} r_{ih\theta p} \tag{32}$$

This constraint alone is not enough to ensure manufacturable composite patches. Three further sets of constraints need to be defined to ensure the blending between individual layers is achieved in a meaningful manner. First of all, a left hand-side ply must blend with only one of the right hand-side plies. References to left and right hand-side plies follow the example illustrated in Fig. 3a.

$$\sum_{p}^{P_h} t_{(i-p+1)h(P_h-p+1)} \leq 1 \quad \forall i, h \tag{33}$$

Secondly, a right hand-side ply must blend with exactly one of the left hand-side plies.

$$\sum_{p}^{P_h} t_{ihp} = 1 \quad \forall i, h \tag{34}$$

Thirdly, in order to avoid unmanufacturable designs that lead to plies intersecting (e.g. right hand-side 1 ply matched with left hand-side 2 and left hand-side 2 matched with right hand-side 1) the following constraints are defined.

$$\sum_{p}^{P_h} t_{(i-1)hp}(P_h - p + 1) \leq \sum_{p}^{P_h} t_{ihp}(P_h - p + 1) \quad \forall i, h \tag{35}$$

Finally, the internal covering ply rule is formulated as:

$$\sum_{j_1}^{j_1+N} \sum_{p}^{P_h} t_{(i-p+1)h(P_h-p+1)} \leq 1 \quad \forall i, h \tag{36}$$

The presented set of equations is implemented using the Python interface of Gurobi.

3.3. Problem decomposition strategy

Both formulations presented in the previous sections solve the stacking and blending optimisation problem considering all patches and constraints simultaneously. In this section, a decomposition strategy is presented for the implicit formulation of the optimisation problem. The reason for choosing implicit instead of explicit is because the strategy can be applied with greater freedom to any patch in the structure. This is due to the lack of ply-voids in the implicit formulation.

The flowchart of the decomposition strategy is presented in Fig. 4. The user needs to define a path or a set of paths containing all of the patches in the structure. Starting from a specific patch, one neighbouring patch is added at a time, until all patches have been included within the path. For each of the paths defined by the user, the first sub-problem to be optimised only includes the first two patches in the path. These two patches are optimised only considering design rules applied on them and manufacturing rules applied on their interface. Once the optimisation of this sub-problem is converged, the next neighbouring patch defined in the path is included in the optimisation sub-problem. This time, the stacking sequences of the previously optimised patches are fixed by using equality constraints. If the optimisation is carried out successfully, the patch which was added last is fixed and the next neighbouring patch is added in the structure along with all the relevant design and manufacturing rules. In cases where the newly added patch borders two or more of the pre-existing patches whose stacking sequence is fixed, the optimisation problem might become infeasible. Consequently, the fixation constraints applied to the patch being the latest addition, are removed. This operation is repeated until a feasible solution can be retrieved by the optimisation. Once all of the patches have been introduced in the optimisation sub-problem, the procedure is applied for the next user-defined path. The decomposition method based on the path which resulted to the stacking sequence with the lowest objective function value, is used to initialise the complete optimisation problem which considers all constraints simultaneously, without the stacking sequence of any of the patches being fixed.

Even if all possible paths are considered in this decomposition strategy, there is no guarantee that the optimal solution will be obtained in the end. A way to improve the quality of this local minimum would be to add two or more patches in each sub-problem at a time, instead of one. This increased overlapping allows the optimisation process to overcome the issue of getting stuck at local minima, since stiffness requirements of more ‘free’ patches would be considered in the objective function of each sub-problem. The drawback of this strategy would be the increased computational time for each sub-problem, since the number of plies free to change and the number of interfaces between the patches increases. It should also be noted that the quality of the solution achieved by the decomposition strategy highly depends on the target stiffnesses provided by the gradient-based optimisation

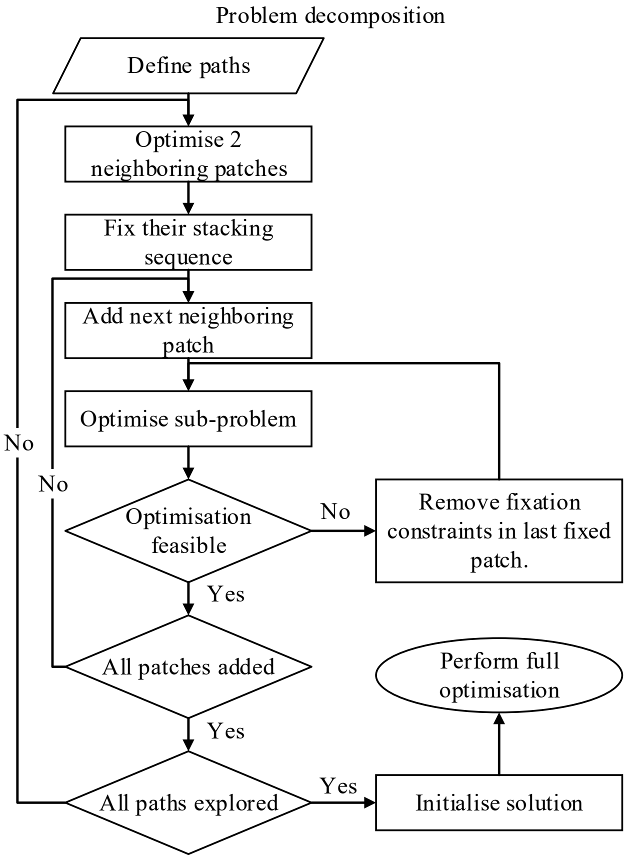

Problem decomposition
Define paths
Optimise 2 neighboring patches
Fix their stacking sequence
Add next neighboring patch
Optimise sub-problem
No
No
Optimisation feasible
No
Remove fixation constraints in last fixed patch.
Yes
All patches added
Perform full optimisation
Yes
All paths explored
Yes
Initialise solution

Figure 4: Flowchart of the optimisation using decomposition developed for the implicit formulation.

algorithm applied in the first stage. The accuracy of the formulation of the composite rules, especially the formulation of the blending rule in the first stage, highly impacts how close the resultant solution obtained from the decomposition strategy will be to the the global optimum.

4. Numerical case studies

Initially, the explicit and implicit formulations are studied using some illustrative cases and then a well studied benchmark problem from the scientific literature is used to assess the proposed discrete optimisation approach. Moreover, the MILP formulations are also compared against two previous approaches.

4.1. Explicit and implicit formulations

Both explicit and implicit formulations for discrete optimisation were able to produce fully blended composite designs satisfying all specified design and blending rules. In this section, we first present the results applying the approaches to some methodology demonstration problems. More specifically, blended stacking sequences that fulfil a prescribed set of design and manufacturing rules are manually defined and the accordant lamination parameters, which are used as target values during the optimisation, are calculated for each individual patch. Therefore the optimal objective function value for this set of problem instances is known *a priori* and is equal to zero. All design and manufacturing rules described in Section 2 are used, with the exception of contiguity, and disorientation which is omitted because grouping is used instead. The fibre orientations for the demonstration cases are limited to $0^o, 90^o, \pm 45^o$. The demonstration problem instances are comprised of a different number of patches (2 or 4), maximum number of layers in the thickest patch (40 or 80) and number of ply drops between adjacent patches (4 or 8). In all demonstration cases, patches are placed sequentially one after the other.

Table 2: Comparison of the MILP size for selected demo problems between the explicit and implicit formulation.

Description			Explicit formulation			Implicit formulation		
Patches	Layers	Ply drops	Constr.	Design var.	Non-zeros (%)	Constr.	Design var.	Non-zeros (%)
2	40	4	868	368	1.3	911	393	1.1
2	40	8	868	368	1.3	1146	492	0.9
2	80	8	1852	772	0.6	2745	1163	0.3
4	80	8	5071	1990	0.2	6823	2789	0.1

In Table 2, the final MILP problem size in terms of the number of design variables, constraints and percentage of non-zero elements in the MILP optimisation matrix is presented for a subset of the cases. The final MILP size does not directly correspond to the size one would theoretically calculate given the problem definition, because Gurobi, like other commercial solvers, applies pre-solve algorithms that might reduce the size of the problem. It can be observed that the size grows significantly for large instances, as expected. Moreover, using the implicit formulation results in slightly larger problem instances. Finally, it is worth noting that an increased number of ply drops increases the problem size of the implicit formulation, while the size of the explicit formulation stays the same. This is due to the fact that the design variables and therefore the constraints of the implicit formulation depend on the number of blending possibilities between neighbouring patches, which is analogous to the number of ply drops.

In Fig. 5, the runtimes of the two approaches for all the different demonstration problem instances are presented. The individual runtimes are depicted using half circles of logarithmically scaled radiuses. More specifically, the maximum radius in Fig. 5 represents a runtime of 442s while the smallest one 0.8s. All runs have been performed on a PC with an Intel Core i5-8250U @1.60GHz (4 cores, 8 threads) CPU. The algorithm is stopped if an objective value less than $10^{-2}\times$(number of patches) is found.

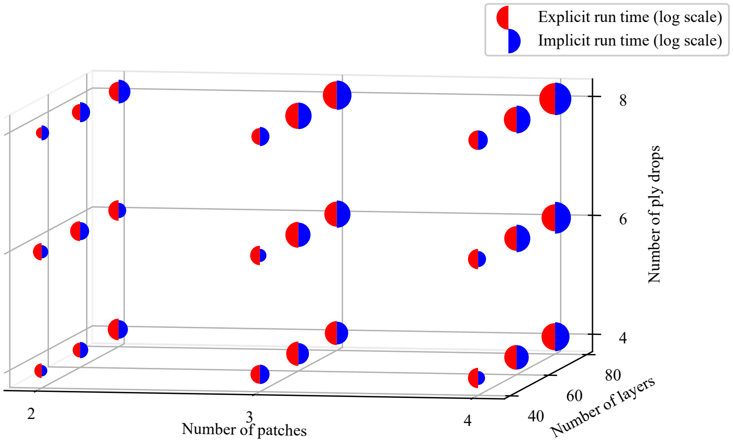


Figure 5: Illustration of run-times for the implicit and explicit formulations for the initial set of problem instances as the number of patches, layers and ply drops change.

Results presented in Fig. 5, highlight the expected increase in run-time as the number of layers and patches increase. The implicit approach is better than the explicit approach for small sized problems, however, as the problem size increases the explicit formulation is either comparable or better than the implicit one in terms of time, as expected. Finally, the runtime of the implicit formulation strongly depends on the number of ply drops between adjacent patches, with more ply drops resulting in higher runtimes, while the explicit formulation does not necessarily depend on the number of ply drops. This can be explained by the fact that more design variables and constraints need to be introduced in the implicit formulation to account for more ply drops.

Fig. 6 illustrates the 18 panel horseshoe benchmark problem. N_x and N_y force resultants are given in lbf/in ($\times 175.1$ to convert into N/m). The modulus of elasticity across the fibre direction is $E_1 = 141\text{GPa}$ and $E_2 = 9.03\text{GPa}$ across the transverse direction, the shear modulus is $G_{12} = 4.27\text{GPa}$ and the principal Poisson's ratio is $\nu_{12} = 0.32$. The ply thickness is 0.191mm.

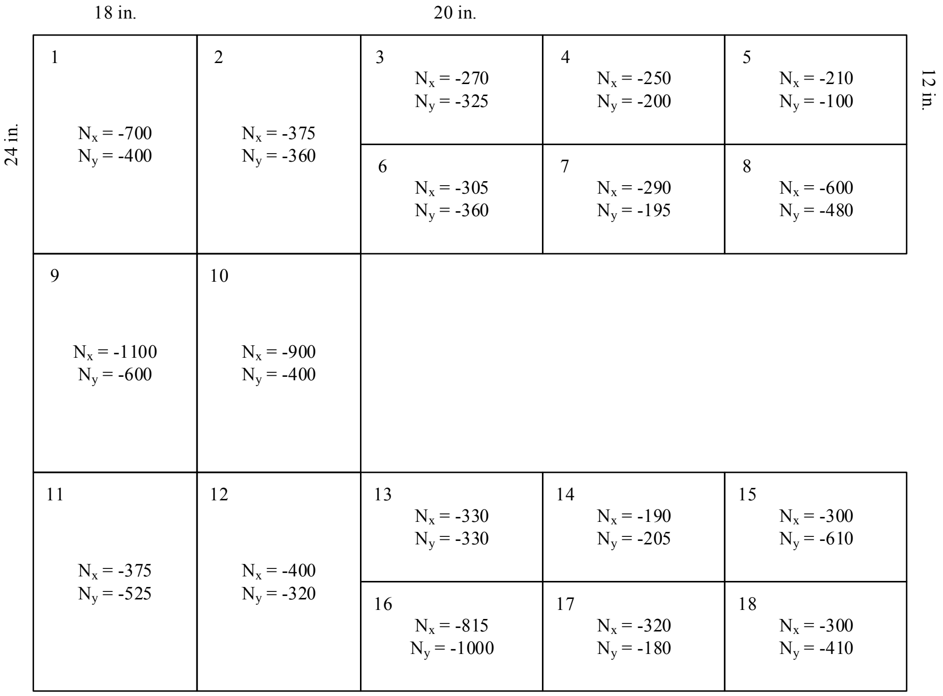


Figure 6: Definition of the 18 panel horseshoe blending problem [43].

In Fig. 7 the convergence plots of the two different MILP formulations are given for the horseshoe problem [43]. The 'best known objective' in Fig. 7 corresponds to the smallest objective function value of a solution satisfying the constraints of the optimisation, while 'best possible objective' refers to the minimum discovered bound for the value of the objective function which is known due to the the branch and bound based algorithm being employed. Optimality of the solution is proven when the value of the best possible objective is equal to that of the best known objective. The optimisation is stopped after 600s because proving optimality becomes time consuming. As observed earlier, the implicit formulation outperforms the explicit one, since the maximum number of layers is rather small for this problem, as shown in Table 3. This is not necessarily the case for larger problems for which the explicit approach can converge to better solutions in less computational time. The disadvantage of the explicit formulation though is the lack of providing a good theoretical bound for the objective (Fig. 7), which in turn results in bigger computational times until proving convergence.

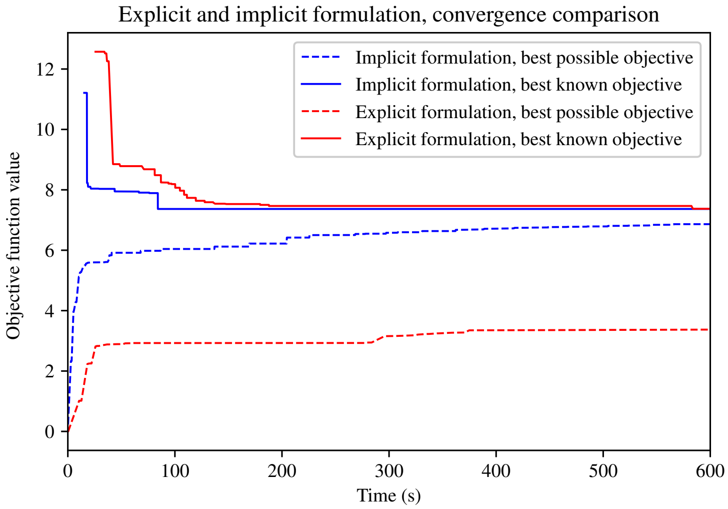


Figure 7: Convergence comparison between the explicit and implicit formulation for the 18 panel horseshoe problem.

Next, a comparison between the convergence of the decomposition strategy and the non-decomposed implicit formulation is performed for the 18

Table 3: Number of layers for the patches of the 18-panel horseshoe problem.

Patch	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Layers	34	28	22	20	16	22	20	26	38	36	30	28	22	20	26	32	20	22

panel horseshoe problem. Three different paths, representing patch sequences, are defined to test the decomposition strategy as shown in Table 4. The first path begins from panel 5 in the top right of the horseshoe pattern and moves counterclockwise to patch number 18 in the bottom right following a specific pattern. The second path is built up starting from the thinnest patch in the structure and by gradually adding the thinnest possible neighbour each time. The third path is constructed by starting from the thickest patch in the structure and by choosing the thickest neighbour to add each time.

Table 4: Paths containing the sequence of patches used for the optimisation using decomposition of the 18 panel horseshoe problem.

Path	Patch sequence
1	[5,8,7,4,3,6,2,1,10,9,11,12,13,14,15,16,17,18]
2	[5,4,3,7,8,6,2,1,10,9,12,11,13,14,17,18,15,16]
3	[9,10,1,2,3,6,4,7,8,5,11,12,16,13,14,17,15,18]

Fig. 8 illustrates the progress plot for the objective value using the approaches with and without decomposition, where the three triangles mark the objective value of each local minimum discovered by using the three different paths. The one used to initialise the optimisation without the decomposition is that of the best found solution which corresponds to the second path. For the specific manufacturing rules used, the average run-time of each individual run for the different paths is approximately 10 seconds, including the overhead of setting up each of the many sub-problems.

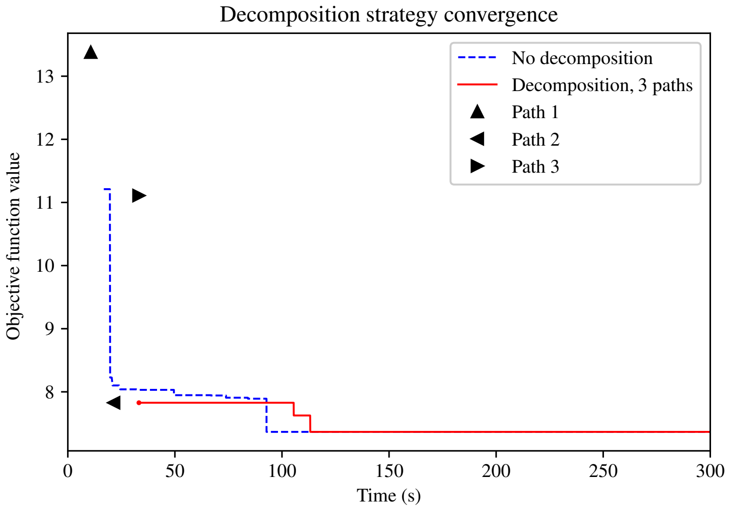


Figure 8: Convergence comparison between the implicit formulation and the decomposition strategy for the 18 panel horseshoe problem.

The fact that the path which resulted to the best solution of the optimisation with the decomposition is the one starting from the thinnest patch and moving to the thinnest neighbour is not incidental. This can be attributed to the fact that in thinner patches, the placement of a layer influences the stiffness more compared to a thicker panel. Therefore, allowing the thinner patches to be optimised first can prevent the decomposition from getting stuck at a local minimum favouring some of the thickest patches. Finally, as for the actual convergence, no significant speed up is observed for the examined case. The optimisation with no decomposition actually converges to what appears to be a global minimum faster than the optimisation approach using decomposition. However, the benefit of using decomposition becomes more evident as the size of a problem grows which is the case for realistic aircraft structures.

To demonstrate this effect, the loads applied to the original formulation of the horseshoe panel problem are multiplied by a factor of 8. The material and dimensions of the panels remain the same. After performing the first optimisation stage again, this increase in loads leads to a doubling of the panels' thicknesses. Fig. 9 depicts the improvement of the value of the objective function with respect to time, for the modified horseshoe problem when applying the optimisation with and without the decomposition technique. For this bigger optimisation problem, applying the optimisation with the decomposition yields a much better solution in a considerably shorter time frame. A second observation is that each of the 3 different paths now requires an average of approximately 30 seconds to optimise compared to 10 in the original problem definition. This is expected, since the number of layers have doubled and the size of the optimisation problem has increased accordingly. As the number of layers increases, the design space of the laminate becomes wider. A wider design space also means that the current thinnest patch in the structure does not influence the quality of the decomposition that much. Indeed, in the this modified problem, the path which results to the lowest objective function is the third one which starts from the thickest patch in the structure. What is more, the deviation between the objective function values of the three paths is now considerably smaller compared to the original. Finally, because the layers increase, the potential to match the same target stiffness increases which leads to a lower global optimum for this modified problem.

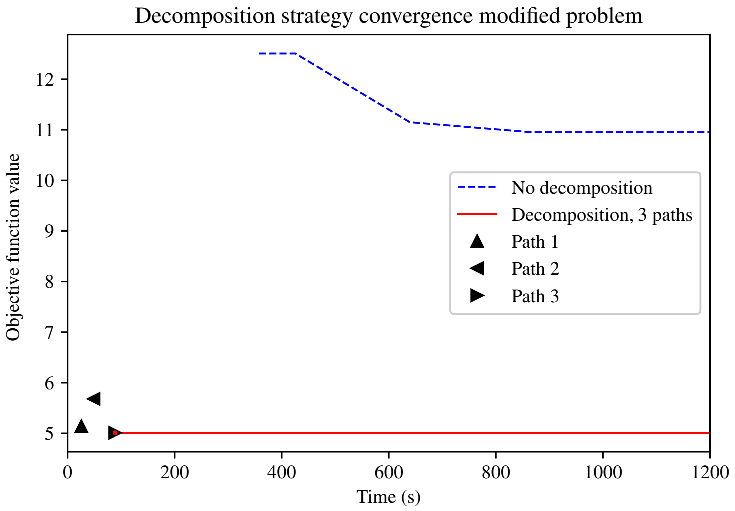


Figure 9: Convergence comparison between the implicit formulation and the decomposition strategy for the modified 18 panel horseshoe problem in which the applied loads are multiplied by a factor of 8.

Overall, both the explicit and implicit formulations provide stacking sequences that are fully compliant with respect to the specified design and manufacturing rules. Based on the observations and experience gained up to now, the explicit formulation can outperform the implicit one for large industrial problems. However, the explicit formulation still lacks the implementation of the disorientation and contiguity constraint as well as the decomposition technique.

4.2. Comparison against relevant work

Liu et al. [29] introduced a logic-based method which is a combination of the branch and bound and global layerwise techniques, developed to retrieve stacking sequences that optimally match some given lamination parameters for single patches. Table 5 compares the results obtained from the presented methodology against the published results for various problems introduced by Liu et al. [29]. Each example uses a different set of design rules when matching the lamination parameters. In Liu et al. [29] two runtimes are provided for each instance; one is the time when the optimum solution (*t.* to opt.) is found and the other one is the time (Total t.) until the search of the tree is completed and hence optimality is proven. The time shown for the implicit MILP formulation of this work is the time spent until optimality is proven. In the overall, using a mathematical programming algorithm to achieve the same task is shown to outperform the logic-based method. This is due to the fact that a search of the entire tree is not necessary to prove optimality and because more sophisticated techniques such as presolving, cutting planes and heuristics are used to assist the optimisation convergence. It should also be noted that for example 1 in Table 5, an objective function with a lower value was calculated using Gurobi. The provided runtimes for the current work have been produced using 1 thread on a 1.6GHz computer compared to a 4GHz computer used in the work of Liu et al. [29].

Finally, the open source genetic algorithm developed by Vicente [48] is used to optimise the horseshoe problem described in the previous section. The genetic algorithm uses the Stacking Sequence Table encoding. All design and manufacturing rules are used, except for the grouping rule, because the disorientation design rule is activated instead. The settings used for the runs of the genetic algorithm are summarised in Table 6. The convergence history of various genetic algorithm runs and of the implicit formulation are presented in Fig. 10. All algorithm runs have been performed on the same

Table 5: Single patch optimisation results, comparison against the work of Liu et al. [29].

	Results from [29]			MILP Implicit formulation	
Example	t. to opt.(s)	Total t.(s)	Obj.	t.(s)	Obj.
1	4.77	3600+	0.0806	3.83	0.0794
2	0.52	7.10	0.1729	0.13	0.1729
10	3.76	3600+	0.0892	1.69	0.0892
11	0.23	5.95	0.0984	0.16	0.0984
13	0.19	0.90	0.3828	0.14	0.3828
14	0.97	1.59	0.3776	0.18	0.3776
15	6.40	9.06	0.1120	0.10	0.1120

1.6GHz CPU using only one core. The implicit formulation converges to a much better solution. This can be attributed to two reasons. First of all, the blending formulation in this work allows for more design freedom, as explained in section 2, which leads to a smaller feasible objective function. Secondly, the mathematical programming algorithm used to solve the problem formulation, as presented in the current work, can consistently lead to the optimal value of the objective function, while the genetic algorithm gets trapped in different local minima, based on the contextual random number generator seed. However, the genetic algorithm is able to provide decent feasible solutions in less time.

5. Conclusions

Both formulations can be used in conjunction with mathematical programming algorithms and are able to retrieve laminated composite structures

Table 6: Settings used in the genetic algorithm of Vicente [48].

Parameter	Value
Population size	200
Maximum number of generations	1000
Elitism percentage	4%
Probability of mutation	15%
Probability of crossover	75%

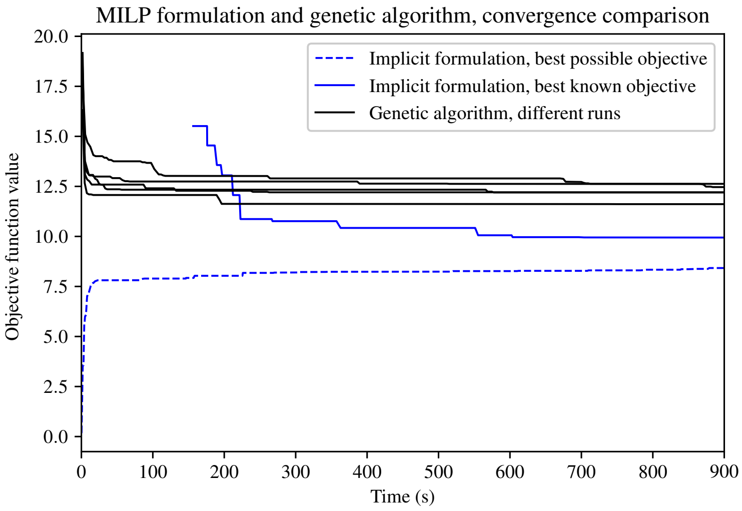


Figure 10: Convergence comparison between the implicit formulation and a guide-based genetic algorithm [48] for the 18 panel horseshoe problem.

which fulfil a set of prescribed design and manufacturing rules. The objective function of this optimisation involves the minimisation of the absolute difference between the stiffness of the computed design and a set of target stiffness characteristics provided by the first stage of the optimisation.

The derived formulations, namely explicit and implicit, mainly differ in the way blending is treated. The explicit formulation offers a much more direct approach to the representation of blending, which however increases the complexity of defining the objective as a linear function of the available design variables. The derivation of blending constraints for the implicit formulation is more complex, however other composite rules and the objective function can be easily derived. Finally, a decomposition technique has been developed for the implicit formulation to assist with the discovery of good local optima in a much shorter time frame.

The results presented show that both formulations are able to tackle a widely used benchmark in a reasonable time frame. The explicit formulation is able to tackle problems of larger size in a shorter time. However, the utilisation of the decomposition technique on the implicit formulation is very promising for applications in industrial composite parts of much larger scales. Compared to an open-source genetic algorithm, the current formulations lead to better discrete stacks in a given time frame. This can be partly attributed to the fact that the current formulations offer more design freedom than most available blending representations and partly to mathematical programming algorithms, offering a more robust convergence towards the global optimum compared to heuristics. Comparison of the current MILP formulation against a custom-made Branch and Bound algorithm focusing on the stacking sequence optimisation of single patches, shows that optimality of the solution can be proven faster for all examined cases.

The present paper offers two novel formulations of the thoroughly examined discrete blending and stacking sequence optimisation problem. Future work may involve the incorporation of manufacturing cost requirements and objectives in the formulation. The explicit formulation might be better suited for such an addition, but the drawbacks of its implementation must first be dealt with. Improvements on the quality of the solution retrieved by the decomposition technique can also be achieved by overlapping more patches for each sub-problem at an extra computational cost, or by exploring more advanced techniques of removing fixation constraints is the case a sub-problem becomes infeasible.

Acknowledgments

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 764650.

6. Declaration of Interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

7. Data availability

The data that support the findings of this study are available on request from the corresponding author.

References

[1] Adams, D. B., Watson, L. T., Gürdal, Z., & Anderson-Cook, C. M. (2004). Genetic algorithm optimization and blending of composite lam-

inates by locally reducing laminate thickness. *Advances in Engineering Software*, *35*, 35 – 43.

[2] Allaire, G., & Delgado, G. (2016). Stacking sequence and shape optimization of laminated composite plates via a level-set method. *Journal of the Mechanics and Physics of Solids*, *97*, 168–196.

[3] An, H., Chen, S., & Huang, H. (2018). Stacking sequence optimization and blending design of laminated composite structures. *Structural and Multidisciplinary Optimization*, .

[4] Aymerich, F., & Serra, M. (2008). Optimization of laminate stacking sequence for maximum buckling load using the ant colony optimization (aco) metaheuristic. *Composites Part A: Applied Science and Manufacturing*, *39*, 262–272.

[5] Bermell-Garcia, P., Verhagen, W. J., Astwood, S., Krishnamurthy, K., Johnson, J. L., Ruiz, D., Scott, G., & Curran, R. (2012). A framework for management of knowledge-based engineering applications as software services: Enabling personalization and codification. *Advanced Engineering Informatics*, *26*, 219 – 230.

[6] Bruyneel, M. (2011). Sfp—a new parameterization based on shape functions for optimal material selection: application to conventional composite plies. *Structural and Multidisciplinary Optimization*, *43*, 17–27.

[7] Bruyneel, M., Zein, S., & Grihon, S. (2014). A two-step optimization approach for the optimal design of composite structures, including geo-

metric non-linear behavior, design rules and manufacturing constraints. In *4th Aircraft Structural Design Conference*.

[8] Campen, J. V., Seresta, O., Abdalla, M., & Gürdal, Z. (2012). General blending definitions for stacking sequence design of composite laminate structures. In *49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*.

[9] Chang, N., Wang, W., Yang, W., & Wang, J. (2010). Ply stacking sequence optimization of composite laminate by permutation discrete particle swarm optimization. *Structural and Multidisciplinary Optimization*, *41*, 179–187.

[10] Fedon, N., Weaver, P. M., Pirrera, A., & Macquart, T. (2021). A method using beam search to design the lay-ups of composite laminates with many plies. *Composites Part C: Open Access*, *4*, 100072.

[11] Fedon, N., Weaver, P. M., Pirrera, A., & Macquart, T. (2021). A repair algorithm for composite laminates to satisfy lay-up design guidelines. *Composite Structures*, *259*, 113448.

[12] Floudas, C. A. (1995). *Nonlinear and Mixed-Integer Optimization: Fundamental and Applications*. New York: Oxford University Press.

[13] Gao, T., Zhang, W., & Duysinx, P. (2012). A bi-value coding parameterization scheme for the discrete optimal orientation design of the composite laminate. *International Journal for Numerical Methods in Engineering*, *91*, 98–114.

[14] Ghiasi, H., Fayazbakhsh, K., Pasini, D., & Lessard, L. (2010). Optimum stacking sequence design of composite materials part ii: Variable stiffness design. *Composite Structures*, *93*, 1–13.

[15] Ghiasi, H., Pasini, D., & Lessard, L. (2009). Optimum stacking sequence design of composite materials part i: Constant stiffness design. *Composite Structures*, *90*, 1–11.

[16] Glover, F. (1975). Improved linear integer programming formulations of nonlinear integer problems. *Management Science*, *22*, 455–460.

[17] Gurobi Optimization, L. (2018). *Gurobi Optimizer Reference Manual*.

[18] Herencia, J., Weaver, P., & Friswell, M. (2008). Initial sizing optimisation of anisotropic composite panels with t-shaped stiffeners. *Thin-Walled Structures*, *46*, 399–412.

[19] IJsselmuiden, S. (2011). *Optimal Design of Variable Stiffness Composite Structures using Lamination Parameters*. Ph.D. thesis TU Delft.

[20] IJsselmuiden, S., Abdalla, M., Seresta, O., & Gürdal, Z. (2009). Multistep blended stacking sequence design of panel assemblies with buckling constraints. *Composites Part B: Engineering*, *40*, 329–336.

[21] Irisarri, F.-X., Lasseigne, A., Leroy, F.-H., & Le Riche, R. (2014). Optimal design of laminated composite structures with ply drops using stacking sequence tables. *Composite Structures*, *107*, 559–569.

[22] Javidrad, F., Nazari, M., & Javidrad, H. (2018). Optimum stacking

sequence design of laminates using a hybrid pso-sa method. *Composite Structures*, *185*, 607–618.

[23] Jing, Z., Fan, X., & Sun, Q. (2014). Global shared-layer blending method for stacking sequence optimization design and blending of composite structures. *Composites Part B: Engineering*, *69*, 181–190.

[24] Kang, L. S., & Blom, A. W. (2016). Optimization of ply orientations for multi-layer composite parts. US Patent 2017/0228473 A1.

[25] Kiyono, C., Silva, E., & Reddy, J. (2017). A novel fiber optimization method based on normal distribution function with continuously varying fiber path. *Composite Structures*, *160*, 503 – 515.

[26] Le Riche, R., & Haftka, R. (1993). Optimization of laminate stacking sequence for buckling load maximization by genetic algorithm. *AIAA Journal*, *31*, 951–956.

[27] Levary, R. R. (1994). Engineering applications of operations research. *European Journal of Operational Research*, *72*, 32 – 42.

[28] Liu, D., Toropov, V., Zhou, M., Barton, D., & Querin, O. (2010). Optimization of blended composite wing panels using smeared stiffness technique and lamination parameters. In *51st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*.

[29] Liu, X., Featherston, C. A., & Kennedy, D. (2019). Two-level layup optimization of composite laminate using lamination parameters. *Composite Structures*, *211*, 337 – 350.

[30] Liu, X., Featherston, C. A., & Kennedy, D. (2020). Buckling optimization of blended composite structures using lamination parameters. *Thin-Walled Structures*, *154*, 106861.

[31] Lund, E., & Stegmann, J. (2005). On structural optimization of composite shell structures using a discrete constitutive parametrization. *Wind Energy*, *8*, 109–124.

[32] Macquart, T., Bordogna, M., Lancelot, P., & De Breuker, R. (2016). Derivation and application of blending constraints in lamination parameter space for composite optimisation. *Composite Structures*, *135*, 224–235.

[33] Meddaikar, Y., Irisarri, F.-X., & Abdalla, M. (2017). Laminate optimization of blended composite structures using a modified shepard's method and stacking sequence tables. *Structural and Multidisciplinary Optimization*, *55*, 535–546.

[34] Montemurro, M., Koutsawa, Y., Belouettar, S., Vincenti, A., & Vannucci, P. (2012). Design of damping properties of hybrid laminates through a global optimisation strategy. *Composite Structures*, *94*, 3309–3320.

[35] Montemurro, M., Pagani, A., Fiordilino, G., Pailhès, J., & Carrera, E. (2018). A general multi-scale two-level optimisation strategy for designing composite stiffened panels. *Composite Structures*, *201*, 968–979.

[36] Montemurro, M., Vincenti, A., & Vannucci, P. (2012). A two-level procedure for the global optimum design of composite modular structures—

application to the design of an aircraft wing part i. *Journal of Optimization Theory and Applications*, *155* , 1–23.

[37] Panettieri, E., Montemurro, M., & Catapano, A. (2019). Blending constraints for composite laminates in polar parameters space. *Composites Part B: Engineering*, *168*, 448 – 457.

[38] Picchi Scardaoni, M., Montemurro, M., Panettieri, E., & Catapano, A. (2020). New blending constraints and a stack-recovery strategy for the multi-scale design of composite laminates. *Structural and Multidisciplinary Optimization*, .

[39] Sanz-Corretge, J. (2017). A procedure to design optimum composite plates using implicit decision trees. *Structural and Multidisciplinary Optimization*, *56* , 1169–1183.

[40] Sanz-Corretge, J., & Echeverría, M. (2018). A novel technique for the design of hybrid composite laminates based on dynamic programming and dynamic tree trimming. *Structural and Multidisciplinary Optimization*, *57*, 1507–1521.

[41] Schuhmacher, G., Daoud, F., Petersson, O., & Wagner, M. (2012). Multidisciplinary airframe design optimisation. In *28th International Congress of the Aeronautical Sciences* (pp. 44–56). volume 1.

[42] Seresta, O., Gürdal, Z., Adams, D., & Watson, L. (2007). Optimal design of composite wing structures with blended laminates. *Composites Part B: Engineering*, *38* , 469–480.

[43] Soremekun, G., Gürdal, Z., Kassapoglou, C., & Toni, D. (2002). Stacking sequence blending of multiple composite laminates using genetic algorithms. *Composite Structures*, *56*, 53–62.

[44] Sørensen, S. N., & Lund, E. (2013). Topology and thickness optimization of laminated composites including manufacturing constraints. *Structural and Multidisciplinary Optimization*, *48*, 249–265.

[45] Stegmann, J., & Lund, E. (2005). Discrete material optimization of general composite shell structures. *International Journal for Numerical Methods in Engineering*, *62*, 2009–2027.

[46] Todoroki, A., & Haftka, R. (1998). Stacking sequence optimization by a genetic algorithm with a new recessive gene like repair strategy. *Composites Part B: Engineering*, *29*, 277–285.

[47] Tsai, S. W., & Pagano, N. J. (1968). Invariant properties of composite materials. In *Composite Materials Workshop*.

[48] Vicente, F. S. (2019). *Stacking Sequence Retrieval of Large Composite Structures in Bi-step Optimization Strategies Using Mechanical Constraints*. Master's thesis TU Delft the Netherlands.

[49] Yang, J., Song, B., Zhong, X., & Jin, P. (2016). Optimal design of blended composite laminate structures using ply drop sequence. *Composite Structures*, *135*, 30–37.

[50] Zein, S., Basso, P., & Grihon, S. (2014). A constraint satisfaction programming approach for computing manufacturable stacking sequences. *Computers and Structures*, *136*, 56–63.

[51] Zeng, J., Huang, Z., Chen, Y., Liu, W., & Chu, S. (2019). A simulated annealing approach for optimizing composite structures blended with multiple stacking sequence tables. *Structural and Multidisciplinary Optimization*, *60*.