Composite Structures 312 (2023) 116853

Contents lists available at ScienceDirect

Composite Structures

journal homepage: www.elsevier.com/locate/compstruct

A novel parallel method for layup optimization of composite structures with ply drop-offs

Xiaoyang Liu [a], Carol A. Featherston [b,*], David Kennedy [b]

[a] School of Civil Engineering, Qingdao University of Technology, Qingdao 266033, China
[b] School of Engineering, Cardiff University, Queen's Buildings, The Parade, Cardiff CF24 3AA, UK

ARTICLE INFO

Keywords:
Composite
Layup optimization
Lamination parameters
Blending
Parallel optimization

ABSTRACT

This paper presents a novel parallel optimization method for obtaining blended layups of composite laminates which closely match target lamination parameters. Firstly, a guide-based adaptive genetic algorithm (GAGA) which stochastically searches the layups is developed. Then the parallel optimization method DLBB-GAGA is developed by combining GAGA and a dummy layerwise branch and bound method (DLBB) which performs a logic-based search in a parallel computation, during which optimization information is shared between the two methods. The combination of these two different methods gives the parallel DLBB-GAGA method the advantages of both, enhancing the searching ability for the blended layup optimization. The superiority of the parallel DLBB-GAGA method is demonstrated through comparisons between the three methods, and it is concluded that the method is particularly effective for practical design where more layup design constraints are considered.

1. Introduction

Because of their excellent mechanical performance, laminated composite structures are increasingly being adopted in various fields of engineering such as aerospace, marine, automotive and civil. In practical design, the ply orientations are usually limited to a set of predefined values (e.g. 0°, 90°, + 45° and − 45°), leading to discrete optimization problems.

In large scale built-up structures, in order to avoid stress concentrations, continuity between stacking sequences in adjacent component panels should be ensured, commonly referred to as the blending problem [1]. By utilizing genetic algorithms (GAs), Liu and Haftka [2] conducted a mass minimisation optimization subject to blending constraints which was implemented by measuring the continuities of layups between adjacent laminates using mathematical expressions. These measurements were treated as general design constraints which were easy to impose on the design variables. The disadvantage of this method, however, was that it could lead to highly constrained problems if the number of design variables was large. Therefore, several blending optimization methods which are able to output blended layups without adding extra constraints have been developed, including methods based on a stochastic search such as the sublaminate-based method [3], guide-based method [4–6], stacking sequence table method (SST) [7,8], shared layer blending method (SLB) [9,10], global shear-layer blending method (GSLB) [11,12], shared-layer and mutation method (SLM) [13], ply drop sequence method (PDS) [14], and a method based on GAs with the use of a ply-composition and a ply-ranking chromosome for each individual. [15]. Although GAs have become the most popular technique for this discrete layup optimization problem, the disadvantages of conducting a random search and the fact that the predetermined parameters have significant influence on optimization performance cannot be ignored. Several alternative methods have therefore been developed, such as the constraint satisfaction programming method (CSP) [16,17] based on a branch and bound tree, the mixed integer linear programming (MILP) method [18], the multi-scale two-level optimization strategy (MS2LOS) [19–21], and topology inspired methods such as the discrete material and thickness optimization method (DMTO) [22,23]. A detailed review of the methods discussed above can be found in [24].

In order to reduce the large number of design variables in layup optimization, lamination parameters [9,10,24–28] and polar parameters [29,30] can be used as design variables in the optimization process. The stiffness matrix can be expressed as a linear function of the lamination parameters instead of the conventional set of equations with large numbers of ply angles. The use of lamination parameters usually divides the optimization into two stages where the continuous optimization of the lamination parameters and laminate thicknesses is

* Corresponding author.
E-mail address: FeatherstonCA@cardiff.ac.uk (C.A. Featherston).

https://doi.org/10.1016/j.compstruct.2023.116853
Received 7 July 2021; Received in revised form 16 January 2023; Accepted 17 February 2023
Available online 18 February 2023

implemented in the first stage and the discrete optimization of layups to match the lamination parameters found follows in the second stage.

As the performance of GAs is mainly dependent on the predefined parameters, efforts have been made to develop adaptive GAs (AGAs) to improve the efficiency of the optimization process. Srinivas and Patnaik [31] first introduced an AGA which uses variable probabilities of crossover and mutation according to the performance of each individual. In the work of Hwang et al. [32], an AGA was utilized for stacking sequence optimization to maximise the natural frequencies of composite laminates. An et al. [33] employed AGAs for the weight optimization of composite laminates under strength and buckling constraints, in their later work [13] an AGA was applied for blending optimization. As parallel computation techniques become more accessible to researchers, parallel computation methods have also been applied to layup optimization to achieve high efficiencies. Punch et al. [34] conducted an optimization using a coarse-grain parallel GA, in which the population was partitioned into several small subpopulations. The optimization was then speeded up by solving small problems, and information was exchanged between these subpopulations, decreasing the probability of premature convergence. In the work of Henderson [35], a master–slave parallel GA was used for layup optimization in which the time-consuming fitness evaluations were parallelized while the selection, crossover and mutation operations were implemented in the main processor for the whole population. Rocha et al. [36] developed a hybrid parallel GA for the optimization of composite laminates. The population was divided into several small subpopulations each of which was assigned different genetic parameters using a coarse-grain parallel GA. A master–slave parallel process was then implemented so that the fitness evaluations of each subpopulation were parallelized to further improve efficiency.

For composite laminates with ply drop-offs, the use of parallel computation methods becomes more necessary. In the work of Adam et al.[4], blended composite laminates were optimized using parallel GAs, with each component panel simultaneously optimized on a different processor. The best individuals from each component panel were exchanged between adjacent populations with local individuals close to the migrants rewarded by giving them higher fitness values, increasing the similarities between adjacent panels. However, fully blended stacking sequences cannot be ensured using this method. In their later work [5], instead of implementing concurrent panel optimizations, a guide-based method using master–slave parallel GAs was developed for blending optimization, with fitness evaluations distributed to different processors to reduce solution time. This parallelization process is particularly beneficial when the fitness values are obtained based on finite element analysis (FEA). In the blending optimizations in [6,37–39], master–slave parallel GAs were utilized to reduce the large amount of computational time incurred by using FEA during the fitness evaluation process. However, whilst parallelization of GAs, splitting one large, time-consuming problem into several small parallel ones can significantly reduce processing time, the inherent drawback of GAs in random searching cannot be overcome.

In the authors' previous work [40], a two-stage layup optimization method based on lamination parameters was developed. In the first stage, the highly efficient software VICONOPT [41,42] based on exact strip analysis and the Wittrick-Williams algorithm [43,44] was used to optimize the lamination parameters and laminate thicknesses with a logic-based layerwise branch and bound method (LBB) employed in the second stage to optimize the layup based on these optimized lamination parameters. This two-stage method was then extended to blending optimization [24], where the lamination parameters and laminate thicknesses of component panels were optimized using the multilevel optimization framework VICONOPT MLO [45,46] in the first stage, and blended stacking sequences were then obtained based on a logical search using the dummy layerwise branch and bound method (DLBB) in the second stage.

The present work focuses on the second stage of this optimization, developing a guide-based adaptive GA (GAGA) which stochastically searches the stacking sequences and then incorporates this into a novel parallel computation method DLBB-GAGA by combining it with the DLBB method. Unlike the parallel methods discussed above, the DLBB-GAGA method combines two types of searches. During the parallel process, GAGA performs a stochastic-based search to optimize the blended stacking sequences, whilst, a logic-based method is performed simultaneously using DLBB. The combination of these two methods gives the parallel method the advantages of both, whilst overcoming the disadvantages of each. Besides, in order to reduce the effect of the predetermined parameters of the GAs on the optimization performance, adaptive probabilities of crossover, mutation and permutation are implemented. The adaptive procedure proposed in this study is further improved by assigning poor individuals higher probabilities of being selected as cross points and mutating to the outer plies. The superior performance of this parallel method is demonstrated through comparisons between these three methods. Section 2 of this paper introduces the lamination parameters. Section 3 describes the process of the two-stage optimization and the newly proposed methods. Numerical results with comparisons between the methods are presented in Section 4. Section 5 provides a brief conclusion.

2. Lamination parameters

The stress-strain relationships in classical laminate theory [47] are presented as:

$$\begin{bmatrix} \boldsymbol{N} \\ \boldsymbol{M} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{B} & \boldsymbol{D} \end{bmatrix} \begin{bmatrix} \boldsymbol{\varepsilon}^0 \\ \boldsymbol{\kappa} \end{bmatrix} \tag{1}$$

where **N** and **M** are vectors of in-plane forces and moments per unit width,**A**, **B** and **D** are the membrane, coupling and bending stiffness matrices, $\boldsymbol{\varepsilon}^0$ is a vector of in-plane strains and κ is a vector of the mid plane curvatures.

The coupling matrix **B** is null for symmetric laminates, and hence will be ignored in this paper. The stiffness matrices **A**and **D** can be expressed in terms of material stiffness invariants **U**, the thickness of the laminate h and 8 lamination parameters ξ_j^k ($j = 1,2,3,4$; $k = A, D$) [48] as:

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

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

The material stiffness invariants U and stiffness properties Q are presented as follows:

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \end{bmatrix} = \frac{1}{8} \begin{bmatrix} 3 & 3 & 2 & 4 \\ 4 & -4 & 0 & 0 \\ 1 & 1 & -2 & -4 \\ 1 & 1 & 6 & -4 \\ 1 & 1 & -2 & 4 \end{bmatrix} \begin{bmatrix} Q_{11} \\ Q_{22} \\ Q_{12} \\ Q_{66} \end{bmatrix} \tag{4}$$

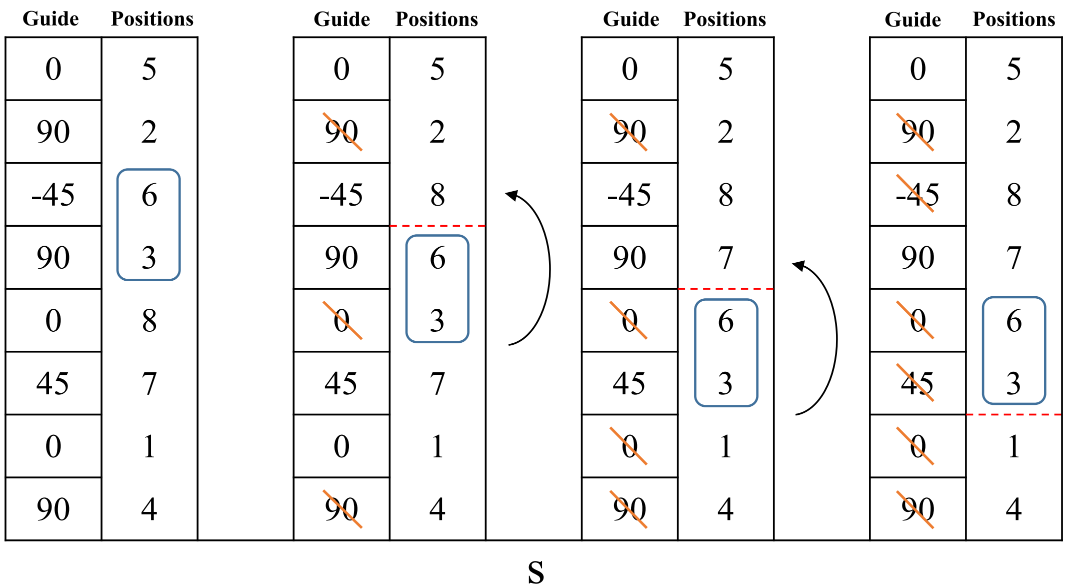

Fig. 1. An illustration example for the implementation of blending constraint in GAGA.

$$\begin{cases} Q_{11} = E_{11}^2/(E_{11} - E_{22}\upsilon_{12}^2) \\ Q_{22} = E_{11}E_{22}/(E_{11} - E_{22}\upsilon_{12}^2) \\ Q_{12} = \upsilon_{12}Q_{22} \\ Q_{66} = G_{12} \end{cases} \tag{5}$$

Where E_{11} is the longitudinal Young's modulus, E_{22} is the transverse Young's modulus, G_{12} is the shear modulus, υ_{12} is the major Poisson's ratio.

The lamination parameters are obtained by the following integrals:

$$\begin{bmatrix} \xi_1^k \\ \xi_2^k \\ \xi_3^k \\ \xi_4^k \end{bmatrix} = \int_{-h/2}^{h/2} Z^k \begin{bmatrix} \cos 2\theta \\ \cos 4\theta \\ \sin 2\theta \\ \sin 4\theta \end{bmatrix} dz, k = A, D, \begin{cases} Z^A = 1/h \\ Z^D = 12z^2/h^3 \end{cases} \tag{6}$$

Where θ represents the ply angle at depth z below the mid-surface. $\xi_4^{A,D}$ are zero when the ply angle is limited to 0°, 90°, $+45^\circ$ and -45°, and ξ_3^A is zero for a balanced laminate.

3. Optimization procedure

Based on the use of lamination parameters, the optimization process is divided into two stages. The aim is to minimize the weight of a blended composite laminate subject to buckling, blending, lamination parameter, and layup design constraints. The lamination parameters and laminate thicknesses of each of the component panels are used as design variables to minimize the weight of the structure in the first stage. Then in the second stage, ply orientations, which are restricted to 0°, 90°, $+45^\circ$ and -45° are used to build the actual blended stacking sequences, based on the layup design constraints to match the optimized lamination parameters obtained in the first stage.

3.1. First stage optimization

For the first stage optimization, the multilevel optimization framework VICONOPT MLO [24,45,46] is used to conduct mass minimization. During the iterative optimization process, FEA is utilized for the static analysis of the whole structure to obtain the load distributions at the start of each design cycle, using stiffness matrices generated based on lamination parameters and laminate thicknesses. The exact strip software VICONOPT is then employed to optimize the lamination parameters and laminate thicknesses of each component panel based on the results from the FEA (stress distribution, geometry, etc), following which the FEA model is updated according to the optimized structural configurations and a new static analysis is implemented to obtain the new load distributions. This process is repeated until convergence criteria on the total mass, individual mass and stress distributions of each panel are reached. Details of this optimization process can be found in [24].

3.2. Second stage optimization

3.2.1. DLBB method

After the optimized lamination parameters are obtained, they are used as targets in the second stage optimization. For the optimization of a single laminate, in the authors' previous work [36] a logic-based LBB method combining a global layerwise technique with the branch and bound method was developed to search stacking sequences to find the one which has lamination parameters closest to the target values subject to layup design constraints (i.e. symmetry, balance, contiguity, disorientation, minimum percentage, and damage tolerance constraints). In order to obtain the layups for large composite structures with ply drop-offs, the DLBB method which incorporates a dummy layerwise technique with the branch and bound method is developed [20]. This DLBB method performs a logical search rather than a stochastic search conducted by a heuristic algorithm. The objective function Γ is obtained by calculating the absolute difference between the target and actual lamination parameters. The dummy layers embedded in the layerwise technique do not contribute to the stiffness but are used to impose the blending constraint. An illustrative example is given in [20] .

3.2.2. Guide-based adaptive GA

Several guide-based methods have been developed and widely used for blending optimization [5,6,14,37,49,50]. The blending procedure in this method is proposed based on the concept of the PDS method [14]. In the GAGA method, each individual comprises two chromosomes with the guide layup which represents the ply angles 0°, 90°, $+45^\circ$ and -45° with 1, 2, 3, and 4, respectively stored in the first chromosome. The second chromosome stores the positions for the plies in the guide layup in a random order. In the optimization, the guide is used as the layup for the thickest panel, and the layups of the thinner ones are obtained by deleting plies from the guide according to the position values from top to

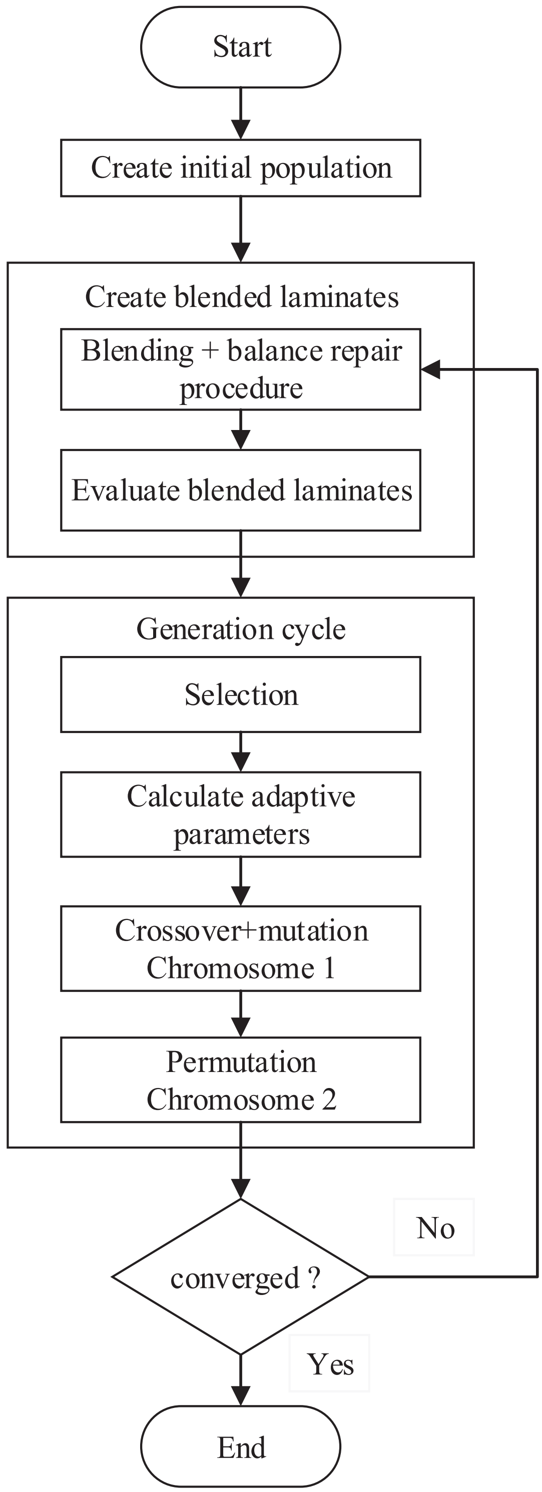

Fig. 2. The flow chart of the GAGA optimization process.

bottom stored in the second chromosome.

In this paper, a repair procedure is developed for implementing the balance constraint. First, the method used to implement the balance constraint proposed in [40] is utilized to repair the guide layup. Then, the second chromosome is repaired using the following procedure: the position values of each − 45° are placed just behind a + 45° ply so that the position values of the + 45° and − 45° plies appear as a set in the second chromosome. During the blending process, if the position value selected for deleting a ply for the generation of a thinner panel is related to a + 45° ply, the next position value related to a 0° or 90° ply below this set is moved so that it is above it, ensuing the number of deleted + 45° and − 45° plies are same. For the case where a thicker panel with an odd number of layers is adjacent to a thinner panel with an even number of layers, the middle layer in the thicker panel should be deleted to satisfy the blending constraint. Hence, the position value of the middle layer in the second chromosome is moved to the top, leading to the middle layer being deleted first.

Fig. 1 shows the GAGA optimization process with an illustrative example, where S denotes a symmetric layup (with a single middle ply if there are an odd number of plies). For each component panel, the left hand side column stores the guide layup and the deleted plies are crossed out, the right hand column gives a random sequence based on which plies are to be deleted, and the plies above the red line have been deleted. The blue box in the right hand column shows a set of + 45° and − 45° plies which must be retained or deleted together to satisfy the balance constraint. In this example, the number of layers for each panel are 16, 9, 8, and 4 respectively, the guide layup is 0/90/− 45/90/0/45/0/90, and the order of the position values is (5 2 6 3 8 7 1 4). The guide layup is used for the thickest panel. For the second thickest panel, the fifth, second and sixth plies in the guide layup are deleted based on the position values in the second chromosome. If the balance constraint is considered, the position value 8 will be inserted above the set of 6 and 3, so the eighth ply will be deleted instead of the sixth ply, ensuring the second panel is balanced. In addition, it should be noted that as the second thickest panel has an odd number of plies, there is just one 0° ply in the middle. When obtaining the layup for the third panel therefore, the position value 7 will be moved to the top of the second chromosome, and the middle layer of the second panel will be dropped off. Finally, the layup of the fourth panel is obtained by deleting a set of + 45° and − 45° plies.

The GAGA optimization process is shown in the flow chart in Fig. 2. As can be seen, the first chromosomes are optimized using two-point crossover and mutation operators, and for the second chromosomes a permutation operator is implemented. The roulette wheel method is utilized in the selection procedure and the elitism operator which keeps the best individuals directly for the next generation is utilized in the optimization process as well. The fitness function in GAGA is represented as:

$$f = \left[\sum_{k=1}^{n} \Gamma_{panel\,k}\right]^{-1} \tag{7}$$

$$\Gamma_{panel\,k} = \sum_{i=1}^{3}\sum_{j=A,D} w_j \left|\xi^{j}_{i(k)(actual)} - \xi^{j}_{i(k)(target)}\right| + \alpha + \beta + \delta + \varepsilon \tag{8}$$

where n is the number of panels in the structure, $\xi^{A,D}_{1,2,3(k)(target)}$ are the target lamination parameters for panel k, and $\xi^{A,D}_{1,2,3(k)(actual)}$ are the actual lamination parameters of the chosen layup of panel k, $w_{A,D}$ are the weighting factors, and α, β, δ, and ε are penalty terms for the layup design constraints.

The performance of standard GAs mainly depends on the predefined crossover and mutation parameters. For stacking sequence optimization, a high probability of crossover P_c with a small probability of mutation P_m is normally used, because frequent crossovers guarantee a random search toward a local or global optimum and mutations are required to prevent the results getting stuck in local optima. Nevertheless, the values of P_c and P_m in standard GAs are left for the user to determine based on empirical experience. In this paper, the values of P_c, P_m as well as the probability of permutation P_p are varied adaptively for each individual and different layers are given different probabilities of being selected as cross points and mutating during the GAGA operations based on their fitness values. The adaptive parameters are obtained by the following equations:

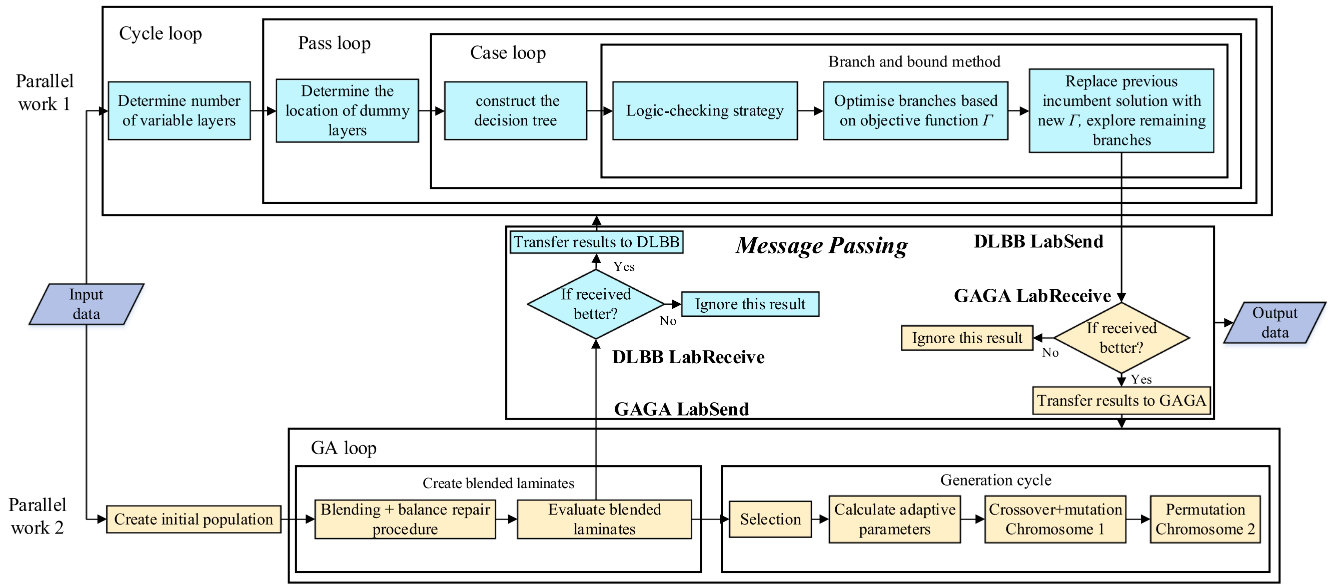

Fig. 3. Parallel optimization process of the DLBB-GAGA method.

$$P_c = \begin{cases} P_{cmax} \bullet \left(\dfrac{f_{max} - f'}{f_{max} - f_{ave}}\right) + P_{cmin} \bullet \left(\dfrac{f' - f_{ave}}{f_{max} - f_{ave}}\right) & f' \geq f_{ave} \\ P_{cmax} & f' < f_{ave} \end{cases} \tag{9}$$

$$P_{cp(i)} = \begin{cases} \frac{1}{n_e} & f' \geq f_{ave} \\ \frac{1}{n_e} + \left(\frac{n_e+1}{2} - i\right) \bullet d & f' < f_{ave} \end{cases} \tag{10}$$

$$P_m = \begin{cases} P_{mmax} \bullet \left(\dfrac{f_{max} - f}{f_{max} - f_{ave}}\right) + P_{mmin} \bullet \left(\dfrac{f - f_{ave}}{f_{max} - f_{ave}}\right) & f \geq f_{ave} \\ P_{mmax} \bullet f_{m(j)} & f < f_{ave} \end{cases} \tag{11}$$

$$f_{m(j)} = Z_u - \left(\frac{j-1}{n_e - 1}\right)(Z_u - Z_l) \tag{12}$$

$$P_p = \begin{cases} P_{pmax} \bullet \left(\dfrac{f_{max} - f}{f_{max} - f_{ave}}\right) + P_{pmin} \bullet \left(\dfrac{f - f_{ave}}{f_{max} - f_{ave}}\right) & f \geq f_{ave} \\ P_{pmax} & f < f_{ave} \end{cases} \tag{13}$$

where P_{cmax}, P_{mmax} and P_{pmax} are the maximum values of P_c, P_m and P_p, respectively; P_{cmin}, P_{mmin} and P_{pmin} are the minimum values of P_c, P_m and P_p, respectively; f is the fitness value of each individual, f' is the larger of the fitness values of the two individuals to be crossed; f_{max} and f_{ave} are the maximum and average fitness values in the population, respectively; $P_{cp(i)}$ are the probabilities of each gene being selected as a cross point; n_e is the number of genes in the chromosome; d is the difference between the values of $P_{cp(i)}$ for adjacent genes; $f_{m(j)}$ is the mutation factor giving different genes different values of P_m; Z_u and Z_l are the upper and lower limits of $f_{m(j)}$; i and j take values from 1 to n_e to calculate $P_{cp(i)}$ and $f_{m(j)}$ from the outermost genes which represent the outermost layers to the innermost ones. In this paper, the parameters discussed above are set with $P_{cmax} = 0.8$, $P_{cmin} = 0.3$, $P_{mmax} = 0.2$, $P_{mmin} = 0.05$, $P_{pmax} = 0.8$, $P_{pmin} = 0.5$, $d = 0.0002$, $Z_u = 1.1$, $Z_l = 0.9$, and the population size is 200.

By using equations (9), (11) and (13), P_c, P_m and P_p increase when all the individuals converge to one layup to prevent premature convergence, and the individuals who have fitness values lower than the average value of the population are implemented with the highest P_c, P_m and P_p to provide sufficient ability to remove poor results, with P_c, P_m and P_p decreasing as individuals' fitness values increase to avoid disrupting the convergence of good results.

In a standard two-point crossover operation, two genes in each parents' chromosomes are randomly selected as cross points, and the genes between these two cross points are swapped between parents to create a new generation. All the genes in the chromosome have the same probability P_{cp} of being selected as cross points. For individuals whose fitness values are lower than average, higher values of P_{cp} are given to the outer layers which have greater contributions to $\xi^D_{1,2,3}$ using equation (10). Based on equation (12), values of P_m are also increased for layers from the innermost to the outermost in these poor individuals. Therefore, for layups which have a poor match with the target lamination parameters, the outer layers are implemented with a higher probability of being selected as cross points and mutating, providing more potential for the improvement of the blending optimization.

3.2.3. Parallel DLBB-GAGA method

As the DLBB method is a logic-based search method which progressively optimizes blended layups from the outer plies to the inner ones, good results can be obtained quickly by solving small problems with small decision trees in the first few cycle loops. However, when the case loop consists of large numbers of plies in later cycle loops, it takes a long time to complete the optimization during which only parts of the laminates are being optimized, limiting the rate of decrease in the value of objective function Γ. As for the stochastic search method GAGA, blended layups at each generation are obtained by randomly deleting plies from the guide layup. Instead of logically implementing the layup design constraints as in the DLBB optimization, GAGA imposes the layup design constraints for each panel using penalty functions, making blended structures with large numbers of component panels easily penalized. In order to combine the advantages of both methods and overcome the disadvantages of each, a parallel DLBB-GAGA method, which conducts the two different methods in parallel computation is developed.

MATLAB is employed for this parallel optimization. The single program multiple data (SPMD) structure in MATLAB makes executing different codes on different cores simultaneously possible, and information can be shared between parallel cores by sending and receiving messages using message passing interface (MPI) based functions such as LabSend and LabReceive.

As can be seen from Fig. 3, the parallel DLBB-GAGA method simultaneously conducts the logic-based DLBB search and the heuristic-based GAGA search on two cores to solve the same blending optimization

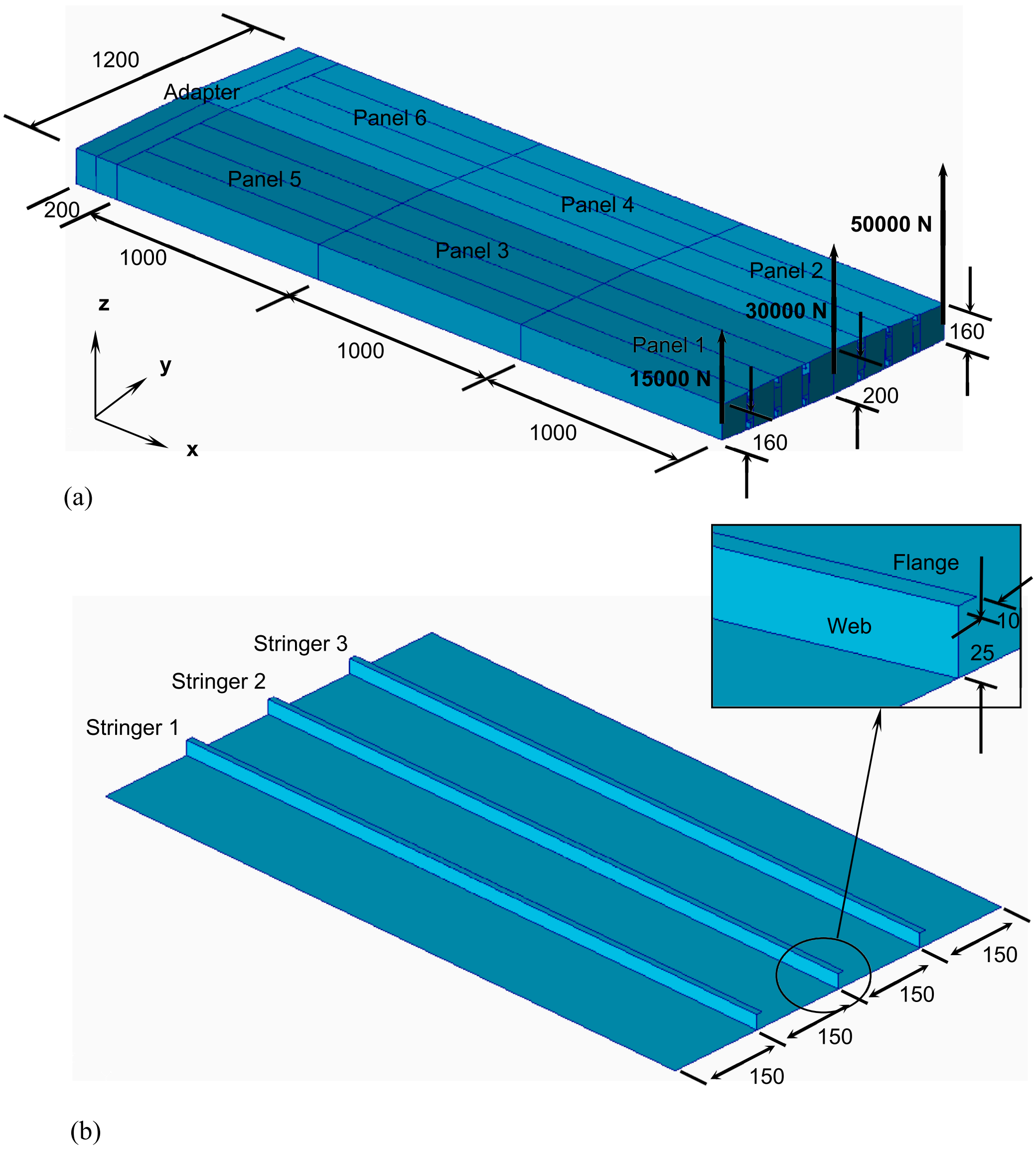

Fig. 4. Details of the wing box structure (dimensions in mm).

problem. Once a new result is obtained by the DLBB search, it is then sent to the GAGA search. If the value of Γ of the received result is smaller than that of the current best result in GAGA, the received result is accepted. After that, the two chromosomes used in GAGA are created based on the received blended layup, and the worst individual in the population is replaced with the received result. GAGA then continues optimizing the blended layup with the updated population. Similarly, when GAGA obtains a new result, the Γ and the layup are sent to DLBB. Once the received result is accepted by DLBB, the dummy layerwise table is updated corresponding to the received blended layup. In addition, if the result from GAGA is received before a case loop, the current Γ, blended layup and dummy layerwise table are replaced with the new result, based on which the optimization of the following case loop is carried on. However, if the result is received during a case loop where the layers to be optimized have been decided in the decision tree, only the Γ is updated and used as a new upper bound during the branch and bound optimization of the current case loop, meaning more branches can be pruned without being explored. The blended layup and dummy layerwise table are replaced with the received result at the end of this case loop, if DLBB has not obtained a better result by itself during the loop.

The benefits of conducting the parallel method are that, at any time both methods are optimizing the blended layups based on the current best result, improving the optimization efficiency. The results received from DLBB bring more diversity to the population of GAGA, speeding up the process of removing local optima result especially when layup design constraints are considered. The disadvantage of the DLBB method in requiring a long time to complete large case loops in later cycles during which only some of the plies in the laminate are allowed to be optimized is also overcome by the parallel process, as any ply in the structure can be optimized by GAGA during that time.

4. Results and discussion

The methods presented herein are proposed for optimizing blended layups to match target lamination parameters as closely as possible. The target lamination parameters provided in [24] are used as targets in this paper to make comparisons between the DLBB method, GAGA and the parallel DLBB-GAGA method. The details of the benchmark wing box used in [24] are shown in Fig. 4, as can be seen each panel has three L-shaped stringers reinforcing longitudinal stiffness. The five separate blending problems in [24] are used herein, including the skins of all of the top panels, the webs of the left hand side panels (i.e. panels 1, 3 and 5) and the right hand side panels (i.e. panels 2, 4 and 6) respectively, and

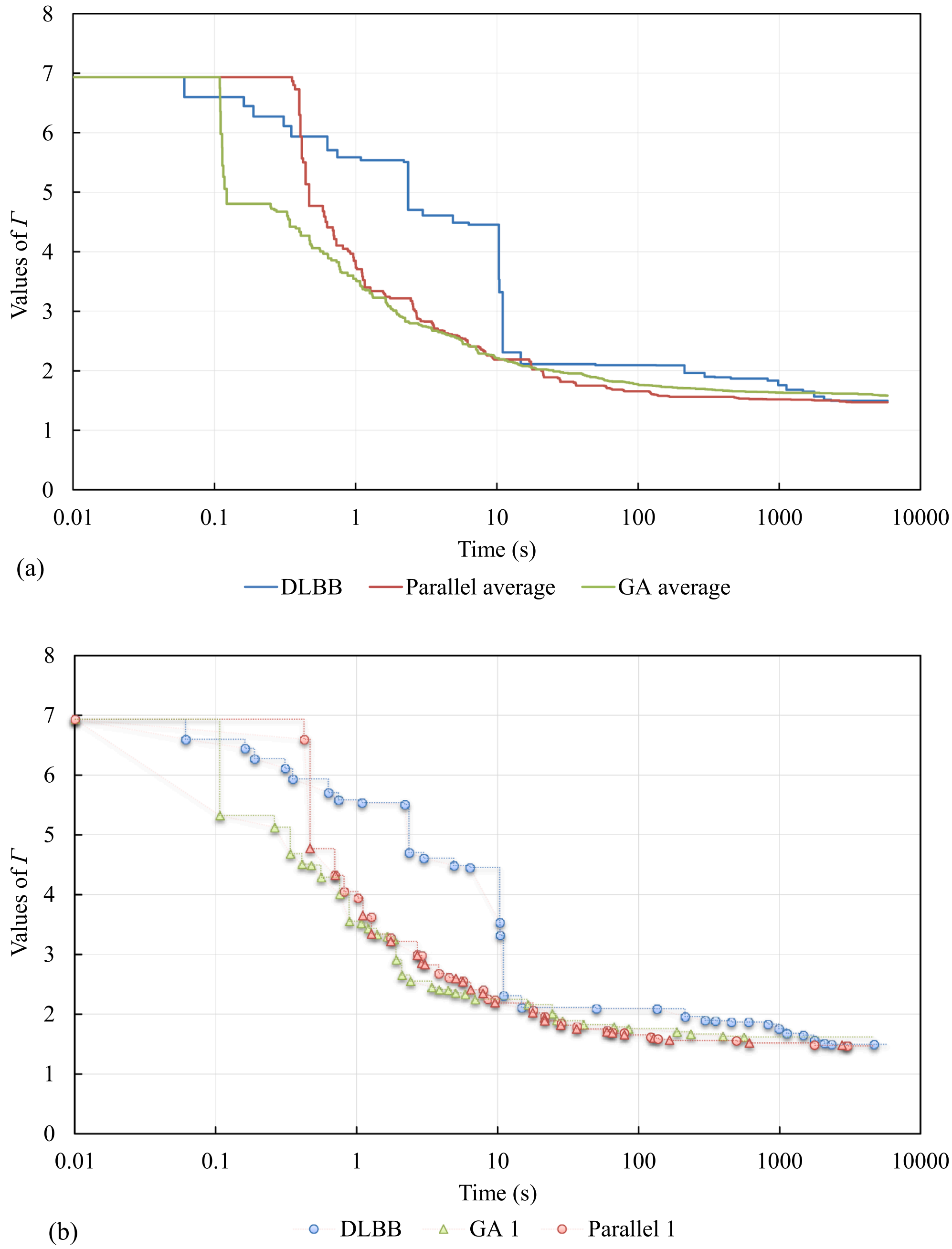

Fig. 5. Comparisons between the DLBB method, GAGA and the parallel method for blended skins. (a) The results of GAGA and the parallel method are averaged values of 10 runs. (b) Examples of a GAGA run and a parallel method run are shown in comparison.

the flanges of the left and right hand panels respectively. There are two sets of results. In the first set, the blended laminates are only required to be symmetric. In the second, the balance constraint as well as the four layup design constraints are imposed to ensure a more practical design. Since GAGA is a stochastic search method, GAGA and the parallel DLBB-GAGA method are run 10 times for each example to guarantee the reliability of the comparison. The same starting layups however are used in all three methods to enable a fair comparison.

4.1. Symmetric case

In the first set of results, the five blending problems in [24] are only restricted to symmetric designs. A comparison between the DLBB method, GAGA and the parallel DLBB-GAGA method for the blended skins is shown in Fig. 5 (a). It can be seen that GAGA finds better results earlier than the other two methods in the early stages. However, after roughly 10 s the differences among the three methods become relatively small and then the parallel method takes the lead until the end. GAGA is more efficient than the DLBB method most of the time, but the final result obtained by the DLBB method is better than that of GAGA. The parallel method achieves the same result as the final result of GAGA after approximately 100 s, and the final result of the parallel method is slightly better than that of the DLBB method, demonstrating its performance in searching blended layups. Note that the parallel method takes longer to obtain its first result because of the overhead of the parallel process. Fig. 5 (b) shows a comparison between DLBB, a GAGA run and a DLBB-GAGA run, the blue circles and green triangles represent the results obtained by the DLBB method and GAGA, individually. The results obtained by the DLBB and GAGA in the parallel method are represented by red circles and red triangles, respectively. It can be seen that both

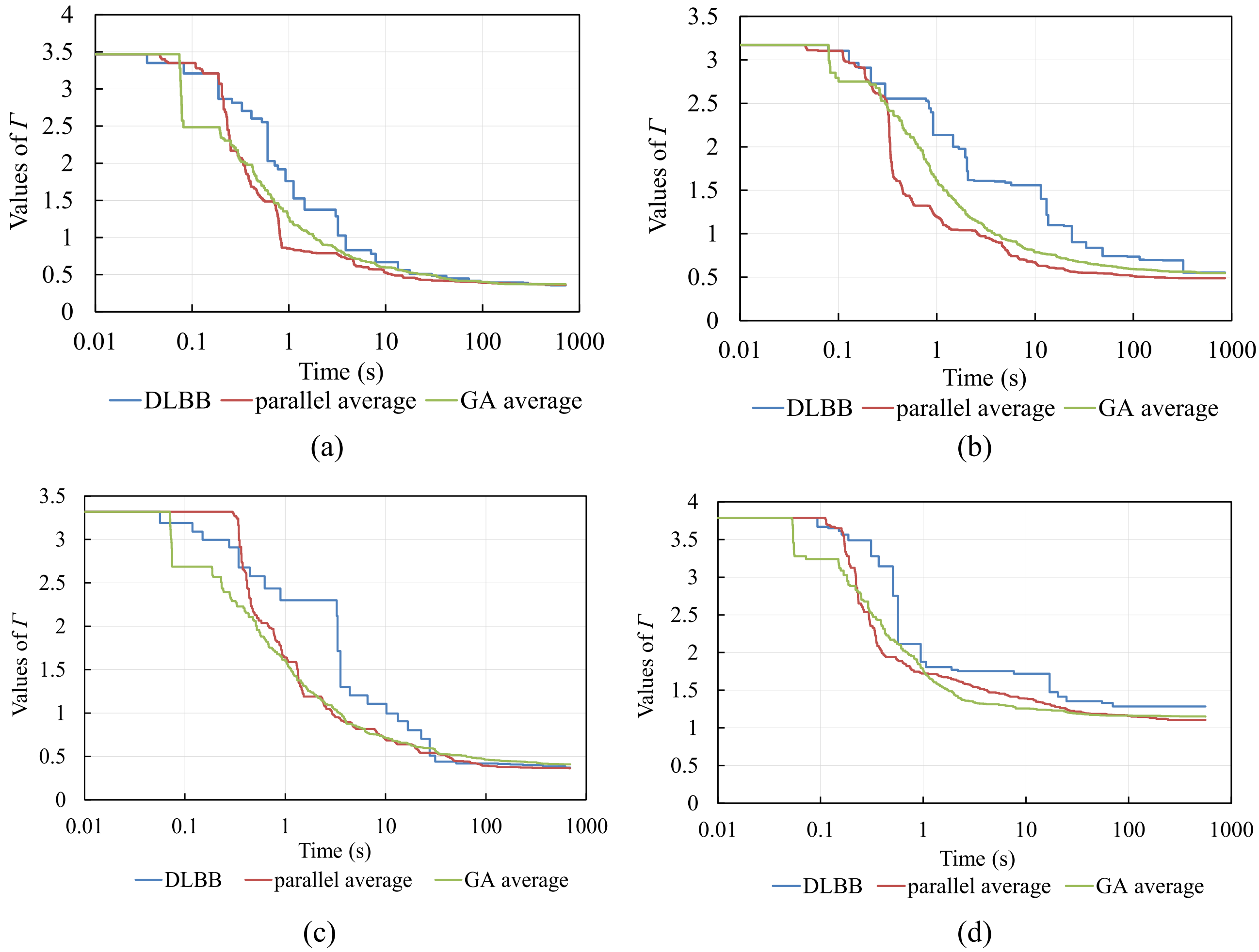

Fig. 6. Comparisons between the DLBB method, GAGA and the parallel method for (a) blended right hand side flanges, (b) blended left hand side flanges, (c) blended right hand side webs, and (d) blended left hand side webs.

methods make significant contributions to the reduction of Γ in the parallel optimization and the message passing between the two methods leads to obvious mutual promotion throughout the whole period of the parallel process, their combination improving the searching capability of each on their own.

For the blended webs and flanges problems which are relatively small, the performances of the three methods are again compared. Comparisons for the right hand side blended flanges are shown in Fig. 6 (a), where it is observed that GAGA performs better than the DLBB method in the first 20 s, after which the results of the two methods are closely aligned until the end. The parallel method finds better results more quickly than the other two from a very early stage in this optimization. Fig. 6 (b) shows the comparisons for the blended left flanges, and it can be seen that the parallel method not only achieves a higher efficiency but also performs better in finding lower values of Γ than each of the two component methods. It only takes the parallel method 10 s to reach the same value of Γ that is achieved by the other two methods at almost 1000 s. As can be seen from Fig. 6 (c), GAGA finds better results earlier than the DLBB method in the first 50 s, after which the DLBB method obtains better results earlier than GAGA. The parallel method performs well during the whole optimization period taking the lead after around 90 s. Fig. 6 (d) shows the comparison for the blended left hand side webs. It is observed that GAGA performs better than the DLBB method in this optimization, and the parallel method surpasses GAGA from approximately 200 s. Thus overall, the parallel method performs better than the other two methods for the blended layup optimization for this set of problems.

4.2. Constrained case

In this set of problems, more practical designs are obtained by enforcing balanced lay-ups and applying four layup design constraints. Fig. 7 (a) presents a comparison between the three different methods for the blended skin problem. As can be seen the values of Γ are larger than those obtained without imposing the extra layup design constraints which narrow the design space. GAGA does not achieve the same final value of Γ as the DLBB method and the parallel method, and the differences between the solutions obtained by the three methods are greater than those obtained without considering these layup constraints. The reason for this is that the constraints are easily violated in the stochastic search of the GA process, especially in the case of a blended structure with several component panels, and because results violating these constraints are penalized in the optimization process, the search capability of GAGA is diminished. This is not the case however for the parallel method; on the contrary, the superiority of the parallel method is more obvious as it leads from the beginning of the optimization with a more distinct advantage. It takes the parallel method around 200 s to achieve the final value of Γ achieved by GAGA in this example. The results support the suggestion that the combination of these two methods overcomes the disadvantages of each and even provides further improvements, making it more appropriate for blended layup optimization.

The comparisons for the right and left hand side flange problems are shown in Fig. 7 (b) and (c), respectively. In contrast to the results obtained without imposing the layup design constraints, the DLBB method

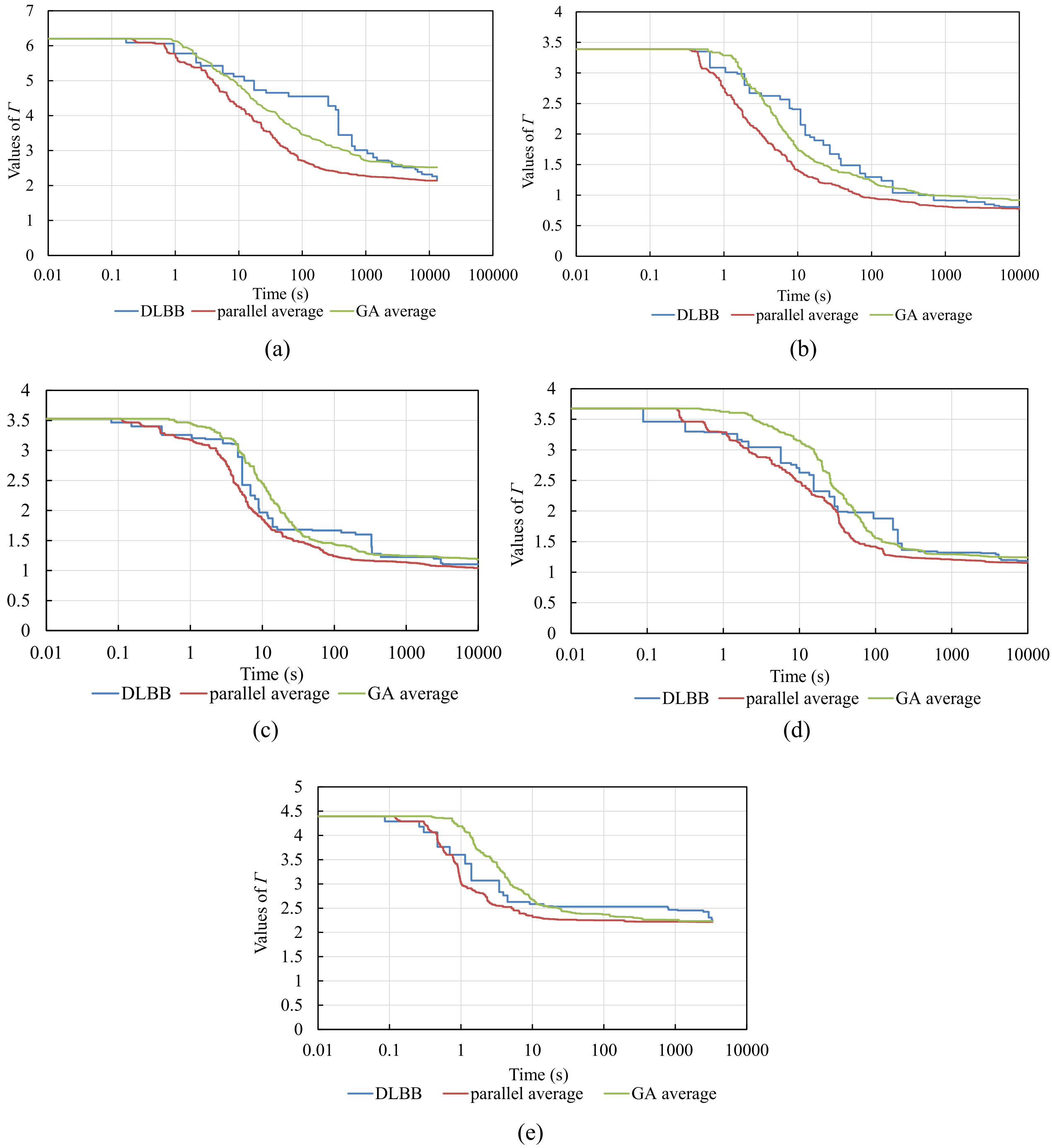

Fig. 7. Comparisons between the DLBB method, GAGA and the parallel method for (a) blended skins, (b) blended right hand side flanges, (c) blended left hand side flanges, (d) blended right hand side webs, and (e) blended left hand side webs, under symmetry, balance, and four layup design constraints.

performs better than GAGA most of the time during the optimizations, and obtains lower final values of Γ than GAGA. With the extra layup design constraints being added, the lead of the parallel method is again more apparent from the start of the optimization. Better still, the parallel method achieves lower final values of Γ than the other two methods. For the right hand side flanges, it takes the DLBB method and GAGA roughly 1,000 and 10,000 s, respectively, to reach the same value of Γ achieved by the parallel method in 100 s. The comparisons for the right hand side web problem are shown in Fig. 7 (d), and it can be seen that after roughly 1 s the parallel method starts to find better results more quickly than the other two methods, and achieves a lower value of Γ at the end. The competition between the DLBB method and GAGA is intense, and in the final stage the DLBB method obtains lower values of Γ than GAGA. The comparisons for the left hand side webs are shown in Fig. 7 (e). In this case the DLBB method performs better than GAGA in the first 10 s, after which GAGA gradually surpasses the DLBB method, but the DLBB method achieves the same value of Γ as the other two methods at a later stage. The parallel method, again, takes a good lead during the optimization, and almost achieves its final value of Γ after just 10 s.

For these more practical designs with additional constraints imposed to ensure manufacturability, the layups obtained at 350 s using the three methods with the dropped plies shown in bold are presented in Tables 1-9. The layups obtained by the DLBB method, a typical GAGA run and a parallel run are listed in Tables 1-3, Tables 4-6 and Tables 7-9,

Table 1
Stacking sequences of skins obtained by DLBB method.

Panel no.	Stacking sequences
5	$[45/-45/0/-45/(90/45_2)_2/90/-45_2/90/45/0/\mathbf{0}/-45_3/90/-45/90_2/45/0_2/45/90/-45_2/(0/45)_2/0_3/45/90_3/-45/0/0]_{\mathrm{MS}}$
6	$[45/-45/0/-45/90/45/\mathbf{45}/90/\mathbf{45/45}/90/-\mathbf{45_2}/90/45/0/-45/-\mathbf{45_2}/90/-45/90_2/45/0/\mathbf{0}/45/90/-45_2/0/45/0/\mathbf{45}/0_3/45/90_2/\mathbf{90}/-45/0/\mathbf{0}]_{\mathrm{MS}}$
4	$[45/-45/\mathbf{0}/-45/90/45/90_2/\mathbf{90}/45/0/(-45/90)_2/\mathbf{90}/45/0/45/90/-45_2/0/45/0_4/45/90_2/-45/0]_{\mathrm{S}}$
3	$[45/-45/-\mathbf{45}/90/\mathbf{45}/90/\mathbf{90/45/0}/-45/\mathbf{90}/-\mathbf{45/90/45}/0/\mathbf{45/90}/-45/-\mathbf{45}/0/45/0_3/\mathbf{0}/45/90_2/-\mathbf{45/0}]_{\mathrm{S}}$
2	$[45/-45/90_2/-45/0/-\mathbf{45}/0/\mathbf{45}/0/\mathbf{0_2}/45/90_2]_{\mathrm{S}}$
1	$[45/-45/90_2/-45/0_3/45/90_2]_{\mathrm{S}}$

Table 2
Stacking sequences of left and right hand side flanges obtained by DLBB method.

Panel no.	Stacking sequences
	left hand side flanges
5	$[45/-45_4/90/45_3/90_2/-45_4/90/45_4/0/45/90/-45/-\mathbf{45}/90/-45/-45/0/45_2/0_2/\mathbf{0}/-45/90_4/45/0_3/\mathbf{45}/0/\mathbf{0}/-45/0_4/45/0/\mathbf{0}]_{\mathrm{MS}}$
3	$[45/-45/-\mathbf{45_3}/90/\mathbf{45_3}/90_2/-45/-\mathbf{45_3/90/45_4}/0/45/\mathbf{(90/-45)_2}/-\mathbf{45/0}/45/\mathbf{45}/0_2/-45/\mathbf{90_4/45/0}/0_3/-45/0_4/45/0]_{\mathrm{S}}$
1	$[45/-45/90_3/-45/0/45_2/0_2/-45/0_3/-45/0_4/45/0]_{\mathrm{MS}}$
	right hand side flanges
6	$[45/-45_4/90/-45_2/\mathbf{90}/-45/90/45_3/90/-45/-\mathbf{45}/0_2/45/90/45_2/\mathbf{45}/0/\mathbf{0}/45/(90/-45)_2/0_3/45/90_3/-45/0_3/45/0_4/45/0/0]_{\mathrm{MS}}$
4	$[45/-45_3/-\mathbf{45}/90/-45/-\mathbf{45_2}/90/45_3/\mathbf{90}/-\mathbf{45}/0_2/\mathbf{45/90/45_2}/0/\mathbf{45/90}/-\mathbf{45/90}/-\mathbf{45}/0_3/45/\mathbf{90_3}/-\mathbf{45}/0_2/\mathbf{0/45/0_4/45/0/0}]_{\mathrm{MS}}$
2	$[45/-45_2/(-45/90)_2/45_3/0_3/-45/0_3/45/0_2]_{\mathrm{S}}$

Table 3
Stacking sequences of left and right hand side webs obtained by DLBB method.

Panel no.	Stacking sequences
	left hand side webs
3	$[45/(-45/90)_2/45/\mathbf{90/45/90_3}/-\mathbf{45}/0_3/-45/0/45/0/\mathbf{0_2/45/0_4}/-\mathbf{45/0/0}]_{\mathrm{MS}}$
1	$[45/-45/90/-45/\mathbf{90/45}/0_2/\mathbf{0}/-\mathbf{45/0}/45/\mathbf{0}]_{\mathrm{MS}}$
5	$[45/-45/90/-45/0_2/45]_{\mathrm{S}}$
	right hand side webs
6	$[45/-45/-\mathbf{45_3}/90_3/\mathbf{45_4}/90/-45/-\mathbf{45_2}/0_2/-\mathbf{45/90/45/90_3}/-\mathbf{45}/0_2/45/0/\mathbf{45}/0_3/-45/90_2/-\mathbf{45}/90_2/\mathbf{90}/45/\mathbf{90_3/45}/0/\mathbf{45}/0/\mathbf{0_3/45/0_4}/-\mathbf{45/90}]_{\mathrm{MS}}$
4	$[45/-45/90/\mathbf{90_3}/-45/0_2/\mathbf{0_2}/45/0_3/\mathbf{0}/-45/90/\mathbf{90_3}/45/0_2]_{\mathrm{S}}$
2	$[45/-45/90/-45/0_2/45/0_3/-45/90/45/0/0]_{\mathrm{MS}}$

Table 4
Stacking sequences of skins obtained by GAGA.

Panel no.	Stacking sequences
5	$[45/-45_3/\mathbf{0}/-45/(90/45_2)_2/45/90_3/-45/0_2/45/90_2/45/0/-45/90_3/-45/0/45/(0/-45)_2/0_2/-45/90/-45/0/45_2/0/0]_{\mathrm{MS}}$
6	$[45/-45_3/-\mathbf{45}/90/45_2/\mathbf{90}/45/\mathbf{45_2}/90_3/-45/0/\mathbf{0}/45/90_2/\mathbf{45/0}/-45/90_2/\mathbf{90}/-45/0/45/0/-\mathbf{45}/0/-\mathbf{45}/0_2/-45/90/-45/0/45_2/0/\mathbf{0}]_{\mathrm{MS}}$
4	$[45/-45_2/-\mathbf{45}/90/45_2/\mathbf{45}/90_3/-45/0/45/(90_2/-45)_2/0/45/0_3/\mathbf{0}/-45/90/-45/0/45_2/0]_{\mathrm{S}}$
3	$[45/-45/-\mathbf{45}/90/\mathbf{45_2}/90/\mathbf{90_2}/-45/0/\mathbf{45/90_2}/-45/\mathbf{90_2}/-\mathbf{45}/0/45/0/\mathbf{0_2}/-45/\mathbf{90}/-\mathbf{45}/0/45_2/0]_{\mathrm{S}}$
2	$[45/-45/90_2/(-45/0)_2/45/\mathbf{0}/-\mathbf{45/0}/45/\mathbf{45}/0]_{\mathrm{S}}$
1	$[45/-45/90_2/(-45/0)_2/45_2/0]_{\mathrm{S}}$

Table 5
Stacking sequences of left and right hand side flanges obtained by GAGA.

Panel no.	Stacking sequences
	left hand side flanges
5	$[45/-45_4/90/-45_2/-\mathbf{45}/90/45_4/90/-45_2/90/45/90_3/-45/0_3/(-45/0)_3/\mathbf{0}/45/90_3/45/0_3/\mathbf{0}/45/\mathbf{45}/90/45/0_3/(45/0)_2/\mathbf{0}]_{\mathrm{MS}}$
3	$[45/-45_2/-\mathbf{45_2}/90/-\mathbf{45_2/90}/45/\mathbf{45_3}/90/-\mathbf{45_2}/90/\mathbf{45}/90/\mathbf{90_2}/-45/0_3/-\mathbf{45}/0/-45/0/-\mathbf{45}/0/\mathbf{45/90_3/45}/0_2/\mathbf{0}/45/\mathbf{90/45}/0/\mathbf{0_2}/45/0/\mathbf{45/0}]_{\mathrm{S}}$
1	$[45/-45_2/90/45/90_3/(-45/0_4)_2/45/0/45/0]_{\mathrm{MS}}$
	right hand side flanges
6	$[45/-45_3/\mathbf{90}/-45/-\mathbf{45}/90/-45/0/(45/90)_2/-45_3/90/45/90/\mathbf{45}/\mathbf{0}/-45/90/45_2/0_2/45/0/-45/90/45/90_3/45/0_3/-45/0_2/45/0_3/45/0/0]_{\mathrm{MS}}$
4	$[45/-45_4/90/-\mathbf{45/0}/45/\mathbf{90}/45/90/-\mathbf{45_3/90}/45/\mathbf{90}/-\mathbf{45/90/45_2}/0_2/\mathbf{45}/0/-45/\mathbf{90/45/90_3/45}/0_3/-\mathbf{45}/0/\mathbf{0}/45/0/\mathbf{0_2/45/0/0}]_{\mathrm{MS}}$
2	$[45/-45_4/90/45_2/90/45/0_3/-45/0_4/45/0]_{\mathrm{S}}$

Table 6
Stacking sequences of left and right hand side webs obtained by GAGA.

Panel no.	Stacking sequences
	left hand side webs
3	$[45/-45/90/\mathbf{45}/90_2/-45/\mathbf{90_4/45}/0/-\mathbf{45_2}/0_3/\mathbf{0}/45/0/-\mathbf{45}/0/\mathbf{0_3}/\mathbf{45/0/0}]_{\mathrm{MS}}$
1	$[45/-45/90/\mathbf{90_2}/-45/0/\mathbf{0_3}/45/0/\mathbf{0}]_{\mathrm{MS}}$
5	$[45/-45/90/-45/0/45/0]_{\mathrm{S}}$
	right hand side webs
6	$[45/-45/-\mathbf{45_3}/90/\mathbf{45}/90/-45/90/\mathbf{45}/90/-\mathbf{45/0/45}/90/\mathbf{90}/45/\mathbf{45}/0/-\mathbf{45/90}/-\mathbf{45/90_2}/-\mathbf{45}/0/\mathbf{45}/0/\mathbf{45/90_4/45/0_3}/-45/\mathbf{90_3/45}/0_2/-\mathbf{45}/0/\mathbf{0_2}/45/90/-45/0_3/45/0/\mathbf{0}]_{\mathrm{MS}}$
4	$[45/-45/90/\mathbf{90}/-45/\mathbf{90_2/90/45}/0_3/-\mathbf{45/0_3}/45/90/-45/0_3/45/0]_{\mathrm{S}}$
2	$[45/-45/90/-45/0_3/45/90/-45/0_3/45/0]_{\mathrm{MS}}$

Table 7
Stacking sequences of skins obtained by parallel method.

Panel no.	Stacking sequences
5	$[45/-45_3/90/45_4/90/-45/0/45_2/\mathbf{0}/45/90/-45_3/90_3/-45/90_2/-45/0_2/45/0/-45/0_3/45/90_3/45/0_2/-45/0/0]_{\mathrm{MS}}$
6	$[45/-45_2/-\mathbf{45}/90/45_3/\mathbf{45}/90/-45/0/\mathbf{45_3}/90/-45_2/-\mathbf{45}/90_3/-\mathbf{45}/90/\mathbf{90}/-45/0_2/45/0/-\mathbf{45}/0_3/45/90_3/45/0_2/-45/0/\mathbf{0}]_{\mathrm{MS}}$
4	$[45/-45/-\mathbf{45}/90/45_2/\mathbf{45}/(90/-45)_2/-45/90_4/-45/0_2/45/0_3/\mathbf{0}/45/90_3/45/0_2/-45/0]_{\mathrm{S}}$
3	$[45/-45/90/\mathbf{45_2}/90/-45/\mathbf{90}/-45/-\mathbf{45/90_4}/-\mathbf{45}/0_2/\mathbf{45}/0_2/\mathbf{0}/45/90/\mathbf{90_2}/45/0_2/-\mathbf{45/0}]_{\mathrm{S}}$
2	$[45/-45/90_2/-45/-\mathbf{45}/0_4/45/90/\mathbf{45/0_2}]_{\mathrm{S}}$
1	$[45/-45/90_2/-45/0_4/45/90]_{\mathrm{S}}$

Table 8
Stacking sequences of left and right hand side flanges obtained by parallel method.

Panel no.	Stacking sequences
	left hand side flanges
5	$[45/-45_4/90/45_4/90/-45_3/-\mathbf{45}/90/45_2/\mathbf{45}/90/-45_2/90_3/45/0/-45/0_2/\mathbf{0_2}/45/90_3/(45/0_3)_2/-45/90/-45/0_4/45/0/\mathbf{0}]_{\mathrm{MS}}$
3	$[45/-45_2/-\mathbf{45_2}/90/\mathbf{45_4}/90/-\mathbf{45_3}/90/45/\mathbf{45/90}/-\mathbf{45_2/90_3/45}/0/-45/0_2/\mathbf{45/90_3/45}/0/\mathbf{0_2}/45/0_3/-45/\mathbf{90}/-\mathbf{45}/0_3/\mathbf{0}/45/0]_{\mathrm{S}}$
1	$[45/-45_2/90_3/45/0/-45/0_3/45/0_3/-45/0_3/45/0]_{\mathrm{MS}}$
	right hand side flanges
6	$[45/-45_4/90/45/90/-45_4/90/45_4/0/-45/90/45/90/\mathbf{90}/-45/90_3/45/0/\mathbf{0_3}/-45/0_4/(45/0_4)_2/45/90/90]_{\mathrm{MS}}$
4	$[45/-45_3/-\mathbf{45}/90/\mathbf{45/90}/-45/-\mathbf{45_3}/90/45_3/\mathbf{45}/0/-\mathbf{45/90/45/90}/-\mathbf{45/90_3/45}/0/-45/0_4/45/0_2/\mathbf{0_2/45/0_4/45/90/90}]_{\mathrm{MS}}$
2	$[45/-45_3/90/-45/90/45_3/0_2/-45/0_4/45/0_2]_{\mathrm{S}}$

Table 9
Stacking sequences of left and right hand side webs obtained by parallel method.

Panel no.	Stacking sequences
	left hand side webs
3	$[45/-45/90/-45/90/45/\mathbf{90_2/-45/90/45/90_2/45}/0_3/\mathbf{0}/-45/0/\mathbf{0}/45/0/\mathbf{0_3/-45/0/0}]_{MS}$
1	$[45/-45/\mathbf{90}/-45/90/45/0_2/\mathbf{0/-45/0/45/0}]_{MS}$
5	$[45/-45_2/90/45/0_2]_S$
	right hand side webs
6	$[45/-45/-\mathbf{45}/90/\mathbf{45}/90/90/-\mathbf{45_2}/90/-45_2/90/45/\mathbf{45_3/90/-45_2/90/45}/0/-\mathbf{45}/0_3/45/90/\mathbf{90_2}/-45/0_4/-\mathbf{45/90_4/-45/90}/45/0_2/\mathbf{0_2/45/90/45/0_4/45/0/0}]_{MS}$
4	$[45/-45/90/\mathbf{90_3}/-45/-\mathbf{45/90/45}/0_2/\mathbf{0_2}/45/90/-45/0_4/45/0/\mathbf{0}]_S$
2	$[45/-45/90/-45/0_2/45/90/-45/0_4/45/0]_{MS}$

respectively. The buckling performance of the obtained stacking sequences are checked using ABAQUS. For the results obtained by the DLBB method, the first and second buckling modes are local buckling modes with buckling load factors equal to 1.06 and 1.11, respectively, with the third buckling mode being a global buckling mode with a buckling load factor equal to 1.15. The same buckling modes occur for the results obtained using GAGA, with buckling load factors of 1.03, 1.13, and 1.15, respectively. As expected, the buckling load factors for the results obtained by the parallel method are slightly higher at 1.08, 1.14, and 1.15, respectively.

These comparisons confirm the advantages of the parallel method in optimizing blended stacking sequences, which are more obvious when imposing the extra layup design constraints. The combination of the logic-based search and stochastic-based searches significantly enhances the searching capability over the whole optimization period, resulting in the value of Γ decreasing until the end. Note that for practical design, the optimization can be terminated as soon as an acceptable result is found (e.g. when the value of Γ of each component panel is less than 0.3).

5. Conclusions

In this paper, a parallel optimization method which simultaneously executes two different methods in a parallel process is developed for the problem of finding a blended layup with lamination parameters that are as close as possible to the target lamination parameters calculated for optimum design. In order to develop the parallel method, a guide-based blending optimization method incorporated within an improved adaptive genetic algorithm is developed. Different probabilities of crossover, mutation and permutation are implemented to different individuals according to their fitness. For individuals whose fitness values are lower than average, higher probabilities of being selected as cross points and mutating are given to the outer plies of the laminate. The resulting stochastic search method GAGA is run in parallel with the logic-based search method DLBB in the parallel DLBB-GAGA method, combining the advantages of both. Comparisons between the DLBB, GAGA and the parallel DLBB-GAGA methods show that benefit is gained from combining the advantages of the different methods, and the parallel DLBB-GAGA method performs with significant superiority in terms of both efficiency and ability of achieving closer matches to the target lamination parameters, especially when extra layup design constraints are imposed in practical design. It should be noted that the parallel computation allows several tasks to be executed simultaneously. In this paper, only two methods are combined together in a parallel process, therefore the advantages of the parallel computation may not be fully utilized. Hence, as a line of future research, the adding of new optimization methods into the parallel method could further improve the optimization performance. Adaptive parameters are applied for crossover, mutation, and permutation operations in this study. Nevertheless, the population size and number of elite in each generation could also affect the optimization performance, hence, applying adaptive procedure for them is also suggested for future work. In addition to this, the results of this paper are obtained based on the use of four ply angles limited to 0°, 90°, $+45^\circ$ and -45°. The performance of the parallel method for larger blending problems considering a larger number of permissible fibre orientations or panels deserves future study.

CRediT authorship contribution statement

Xiaoyang Liu: Conceptualization, Methodology, Software, Validation, Writing – original draft. **Carol A. Featherston:** Writing – review & editing, Supervision. **David Kennedy:** Writing – review & editing, Supervision.

Declaration of Competing Interest

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.

Data availability

Data will be made available on request.

References

[1] Kristinsdottir BP, Zabinsky ZB, Tuttle ME, Neogi S. Optimal design of large composite panels with varying loads. Compos Struct 2001;51:93–102. https://doi.org/10.1016/S0263-8223(00)00128-8.
[2] Liu B, Haftka R. Composite wing structural design optimization with continuity constraints. In: 19th AIAA Appl. Aerodyn. Conf., Anaheim, CA: American Institute of Aeronautics and Astronautics; 2001. https://doi.org/10.2514/6.2001-1205.
[3] Soremekun G, Gürdal Z, Kassapoglou C, Toni D. Stacking sequence blending of multiple composite laminates using genetic algorithms. Compos Struct 2002;56:53–62. https://doi.org/10.1016/S0263-8223(01)00185-4.
[4] Adams DB, Watson LT, Gürdal Z. Optimization and Blending of Composite Laminates Using Genetic Algorithms with Migration Optimization and Blending of Composite Laminates. Mech Adv Mater Struct 2003;10:37–41. https://doi.org/10.1080/15376490306741.
[5] Adams DB, Watson LT, Gürdal Z, Anderson-Cook CM. Genetic algorithm optimization and blending of composite laminates by locally reducing laminate thickness. Adv Eng Softw 2004;35:35–43. https://doi.org/10.1016/j.advengsoft.2003.09.001.
[6] Adams DB, Watson LT, Seresta O, Gürdal Z. Global/local iteration for blended composite laminate panel structure optimization subproblems. Mech Adv Mater Struct 2007;14:139–50. https://doi.org/10.1080/15376490600719212.
[7] Zeng J, Huang Z, Chen Y, Liu W, Chu S. A simulated annealing approach for optimizing composite structures blended with multiple stacking sequence tables. Struct Multidiscip Optim 2019;60:537–63. https://doi.org/10.1007/s00158-019-02223-9.
[8] Irisarri FX, Lasseigne A, Leroy FH, Le Riche R. Optimal design of laminated composite structures with ply drops using stacking sequence tables. Compos Struct 2014;107:559–69. https://doi.org/10.1016/j.compstruct.2013.08.030.
[9] Liu D, Toropov VV, Querin OM, Barton DC. Bilevel Optimization of Blended Composite Wing Panels. J Aircr 2011;48:107–18. https://doi.org/10.2514/1.C000261.
[10] Liu D, Toropov VV, Barton DC, Querin OM. Weight and mechanical performance optimization of blended composite wing panels using lamination parameters. Struct Multidiscip Optim 2015;52:549–62. https://doi.org/10.1007/s00158-015-1244-x.
[11] Jing Z, Chen J, Sun Q. Constrained-manufacturable stacking sequence design optimization using an improved global shared-layer blending method and its 98-line Matlab code. Struct Multidiscip Optim 2019;59:539–75. https://doi.org/10.1007/s00158-018-2083-3.
[12] Jing Z, Fan X, Sun Q. Global shared-layer blending method for stacking sequence optimization design and blending of composite structures. Compos Part B Eng 2015;69:181–90. https://doi.org/10.1016/j.compositesb.2014.09.039.
[13] An H, Chen S, Huang H. Stacking sequence optimization and blending design of laminated composite structures. Struct Multidiscip Optim 2019:59. https://doi.org/10.1007/s00158-018-2158-1.
[14] Yang J, Song B, Zhong X, Jin P. Optimal design of blended composite laminate structures using ply drop sequence. Compos Struct 2016;135:30–7. https://doi.org/10.1016/j.compstruct.2015.08.101.
[15] Fan HT, Wang H, Chen XH. An optimization method for composite structures with ply-drops. Compos Struct 2016;136:650–61. https://doi.org/10.1016/j.compstruct.2015.11.003.
[16] Sanz-Corretge J. A constraint satisfaction problem algorithm for large-scale multi-panel composite structures. Struct Multidiscip Optim 2019;60:2035–51. https://doi.org/10.1007/s00158-019-02309-4.

[17] Zein S, Basso P, Grihon S. A constraint satisfaction programming approach for computing manufacturable stacking sequences. Comput Struct 2014;136:56–63. https://doi.org/10.1016/j.compstruc.2014.01.016.

[18] Ntourmas G, Glock F, Daoud F, Schuhmacher G, Chronopoulos D, Özcan E. Mixed Integer Linear Programming formulations of the stacking sequence and blending optimisation of composite structures. Compos Struct 2021;264:113660.

[19] Picchi Scardaoni M, Montemurro M, Panettieri E, Catapano A. New blending constraints and a stack-recovery strategy for the multi-scale design of composite laminates. Struct Multidiscip Optim 2021;63:741–66.

[20] Scardaoni MP, Izzi MI, Montemurro M, Panettieri E, Cipolla V, Binante V. Multi-scale deterministic optimisation of blended composite structures: case study of a box-wing. Thin-Walled Struct 2022;170:108521.

[21] Scardaoni MP, Montemurro M. A general global-local modelling framework for the deterministic optimisation of composite structures. Struct Multidiscip Optim 2020; 62:1927–49. https://doi.org/10.1007/s00158-020-02586-4.

[22] Sørensen SN, Sørensen R, Lund E. DMTO - A method for Discrete Material and Thickness Optimization of laminated composite structures. Struct Multidiscip Optim 2014;50:25–47. https://doi.org/10.1007/s00158-014-1047-5.

[23] Sjølund JH, Peeters D, Lund E. A new thickness parameterization for Discrete Material and Thickness Optimization. Struct Multidiscip Optim 2018;58:1885–97. https://doi.org/10.1007/s00158-018-2093-1.

[24] Liu X, Featherston CA, Kennedy D. Buckling optimization of blended composite structures using lamination parameters.pdf. Thin-Walled Struct 2020;154:1–13. https://doi.org/org/10.1016/j.tws.2020.106861.

[25] IJsselmuiden ST, Abdalla MM, Seresta O, Gürdal Z. Multi-step blended stacking sequence design of panel assemblies with buckling constraints. Compos Part B Eng 2009;40:329–36. https://doi.org/10.1016/j.compositesb.2008.12.002.

[26] Bordogna MT, Lancelot P, Bettebghor D, De Breuker R. Static and dynamic aeroelastic tailoring with composite blending and manoeuvre load alleviation. Struct Multidiscip Optim 2020. https://doi.org/10.1007/s00158-019-02446-w.

[27] Meddaikar YM, Irisarri FX, Abdalla MM. Laminate optimization of blended composite structures using a modified Shepard's method and stacking sequence tables. Struct Multidiscip Optim 2017;55:535–46. https://doi.org/10.1007/s00158-016-1508-0.

[28] Macquart T, Werter N, De Breuker R. Aeroelastic Design of Blended Composite Structures Using Lamination Parameters. J Aircr 2017;54:561–71. https://doi.org/10.2514/1.C033859.

[29] Panettieri E, Montemurro M, Anita C. Blending constraints for composite laminates in polar parameters space.pdf. Compos Part B Eng 2019;168:448–57. https://doi.org/10.1016/j.compositesb.2019.03.040.

[30] Montemurro M. The polar analysis of the Third-order Shear Deformation Theory of laminates. Compos Struct 2015;131:775–89. https://doi.org/10.1016/j.compstruct.2015.06.016.

[31] Srinivas M, Patnaik LM. Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms. IEEE Trans Syst Man Cybern 1994;24:656–67. https://doi.org/10.1109/21.286385.

[32] Hwang SF, Hsu YC, Chen Y. A genetic algorithm for the optimization of fiber angles in composite laminates. J Mech Sci Technol 2014;28:3163–9. https://doi.org/10.1007/s12206-014-0725-y.

[33] An H, Chen S, Huang H. Laminate stacking sequence optimization with strength constraints using two-level approximations and adaptive genetic algorithm. Struct Multidiscip Optim 2015;51:903–18. https://doi.org/10.1007/s00158-014-1181-0.

[34] Punch WF, Averill RC, Goodman ED, Lin SC, Ding Y, Yip YC. Optimal design of laminated composite structures using coarse-grain parallel genetic algorithms. Comput Syst Eng 1994;5:415–23. https://doi.org/10.1016/0956-0521(94)90023-X.

[35] Henderson JL. Laminated plate design using genetic algorithms and parallel processing. Comput Syst Eng 1994;5:441–53. https://doi.org/10.1016/0956-0521(94)90025-6.

[36] Rocha IBCM, Parente E, Melo AMC. A hybrid shared/distributed memory parallel genetic algorithm for optimization of laminate composites. Compos Struct 2014; 107:288–97. https://doi.org/10.1016/j.compstruct.2013.07.049.

[37] Seresta O, Gürdal Z, Adams DB, Watson LT. Optimal design of composite wing structures with blended laminates. Compos Part B Eng 2007;38:469–80. https://doi.org/10.1016/j.compositesb.2006.08.005.

[38] Jin P, Song B, Zhong X. Structure Optimization of Large Composite Wing Box with Parallel Genetic Algorithm. J Aircr 2011;48:2–5. https://doi.org/10.2514/1.C031493.

[39] Jin P, Zhong X, Yang J, Sun Z. Blending design of composite panels with lamination parameters. Aeronaut J 2016;120:1710–25. https://doi.org/10.1017/aer.2016.88.

[40] Liu X, Featherston CA, Kennedy D. Two-level layup optimization of composite laminate using lamination parameters. Compos Struct 2019;211:337–50. https://doi.org/10.1016/j.compstruct.2018.12.054.

[41] Butler R, Williams FW. Optimum design using VICONOPT, a buckling and strength constraint program for prismatic assemblies of anisotropic plates. Comput Struct 1992;43:699–708. https://doi.org/10.1016/0045-7949(92)90511-W.

[42] Kennedy D, Park B, Unsworth MD. Towards Global Layup Optimization of Composite Panels with Initial Buckling Constraints. proceeding 8th ASMO UK/ISSMO Conf. London 2010:221–31.

[43] Wittrick WH, Williams FW. A general algorithm for computing natural frequencies of elastic structures. Q J Mech Appl Math 1971;24:263–84. https://doi.org/10.1093/qjmam/24.3.263.

[44] Wittrick WH, Williams FW. An algorithm for computing critical buckling loads of elastic structures. J Struct Mech 1973;1:497–518. https://doi.org/10.1080/03601217308905354.

[45] Fischer M, Kennedy D, Featherston C a. Multilevel framework for optimization of lightweight structures. Proc Inst Mech Eng Part G J Aerosp Eng 2012;226:380–94. https://doi.org/10.1177/0954410011411637.

[46] Qu S, Kennedy D, Featherston CA. A multilevel framework for optimization of an aircraft wing incorporating postbuckling effects. Proc Inst Mech Eng Part G J Aerosp Eng 2011;226:830–45. https://doi.org/10.1177/0954410011415158.

[47] Jones RM. Mechanics Of Composite Materials. second edi. New York: Taylor and Francis; 1999.

[48] Diaconu CG, Sekine H. Layup Optimization for Buckling of Laminated Composite Shells with Restricted Layer Angles. AIAA J 2004;42:2153–63. https://doi.org/10.2514/1.931.

[49] Shrivastava S, Mohite PM, Limaye MD. Optimal design of fighter aircraft wing panels laminates under multi-load case environment by ply-drop and ply-migrations. Compos Struct 2019;207:909–22. https://doi.org/10.1016/j.compstruct.2018.09.004.

[50] Albanesi A, Bre F, Fachinotti V, Gebhardt C. Simultaneous ply-order, ply-number and ply-drop optimization of laminate wind turbine blades using the inverse finite element method. Compos Struct 2018;184:894–903. https://doi.org/10.1016/j.compstruct.2017.10.051.