



Improved genetic algorithm for multidisciplinary optimization of composite laminates

Chung Hae Park^{a,*}, Woo Il Lee^b, Woo Suck Han^c, Alain Vautrin^c

^a Laboratoires d'Ondes et Milieux Complexes, FRE 3102-CNRS, Université du Havre, 53 rue de Prony, BP 540, 76058 Le Havre, France

^b School of Mechanical and Aerospace Engineering, Seoul National University, Shinlim-dong, Kwanak-gu, 151-742 Seoul, South Korea

^c Mechanical and Materials Engineering Department, Ecole des Mines de Saint-Etienne, 158 Cours Fauriel, 42023 Saint-Etienne, France

ARTICLE INFO

Article history:

Received 12 October 2007

Accepted 1 March 2008

Available online 12 May 2008

Keywords:

Genetic algorithm (GA)

Multidisciplinary optimization

Memory

Permutation

Local learning

ABSTRACT

We suggest new approaches to reduce the number of fitness function evaluations in genetic algorithms (GAs) applied to multidisciplinary optimization of composite laminates. In the stacking sequence design of laminated structures, the design criteria are classified into two groups, which are layer combination dependent criteria and layer sequence dependent criteria. The memory approach is employed to lessen the number of fitness function evaluations for the identical design individuals that appear during the search. The permutation operator with local learning or random shuffling is applied to the same design individual to improve the fitness for layer sequence dependent criterion, while maintaining the same performance for layer combination dependent criterion. The numerical efficiency of the present method is validated by the sample problem of weight minimization of composite laminated plate under multiple design constraints.

© 2008 Elsevier Ltd. All rights reserved.

1. Introduction

Genetic algorithms have been widely applied to laminate design optimization problems, since they are suitable for integer programming and are more likely to find global optima [1–10]. While they do not require gradient or sensitivity coefficient evaluations, a large number of analyses are, however, required to obtain the optimal solution.

To save computing efforts without losing the reliability of optimization, some attempts have been suggested that involve more than simply adjusting genetic parameters such as population size, selection pressure and the probability of crossover and mutation. One of these attempts is to adopt the memory approach to refer the information that has already been analyzed [8,9]. In the evolutionary search, the same design individual, the fitness of which has been already evaluated, may appear. If the fitness of this identical design is stored in the memory and retrieved for the next search procedure, the total number of fitness function evaluations can be reduced. To store this information in memory, the hashing algorithm and the binary tree algorithm have been employed to construct a data structure for efficient storage and retrieval [11]. The hashing algorithm is, in general, more efficient than the binary tree algorithm [11]. Regardless of its efficiency, however, it is problematic to search for a sufficient hashing function. Otherwise, the bin-

ary tree algorithm has been applied to layer stacking sequence design of laminates [8,9]. In some works, the binary tree memory approach has been proven to work well even for the mixed discrete-continuous optimization problem [9].

Some researchers suggested the Lamarckian co-adaptation of learning and evolution as an effort to improve the reliability and the efficiency of the optimization scheme [12,13]. Learning performs a local search of design parameters, but evolution performs a global search of design parameters among a population. Evolution changes the genotype of an organism at the population level, while learning changes the behavior of an organism at the individual level. Although evolution and learning are two distinct types of changes that occur in two distinct kinds of entities (population and individual organism, respectively), they may influence each other. One hypothesis on how learning influences evolution is the Lamarckian hypothesis that phenotypic traits acquired by an organism during its lifetime are transcribed into the heritable genotype and can be passed on directly to the organism's offspring [14]. According to the viewpoint of modern biology, the Lamarckian hypothesis is no longer accepted as true. Very rarely, otherwise, several intelligent animals, such as human beings, teach their offspring the behavioral patterns acquired during their lifetime, in the form of Lamarckian evolution coded in their brain rather than in their genes. From the viewpoint of computer science, Lamarckian evolution, due to its rapid convergence, can be easy to implement as well as potentially far more effective for solving optimization problems.

* Corresponding author. Tel.: +33 (0) 2 35 21 71 21; fax: +33 (0) 2 35 21 71 99.
E-mail address: chung-hae.park@univ-lehavre.fr (C.H. Park).

In laminate design optimizations, a hybrid genetic algorithm to combine local improvement with global combinatorial search was applied to decrease the computational cost [10]. Response surface approximation with the aid of binary tree memory has been successfully incorporated into the genetic algorithm to locally improve the performance of design individual, and, eventually, the efficiency of optimization was enhanced [10].

Some researchers have introduced new genetic operators, such as “permutation” and “gene addition/deletion,” to augment the standard genetic operators, such as “crossover” and “mutation” [1]. Permutation is a gene arrangement operation that maintains the same gene combination. Some permutation operators are considered to produce a random walk: “insert”, “swap”, “inversion”, and “scramble”. In stacking sequence optimization problems, “gene swap” and “inversion” permutations have been adopted as stochastic operators, as “mutation” is used in the standard genetic algorithm [2,5–7].

In some optimization problems, the decision of permutation itself is the main objective. Representative examples of permutation optimization are the traveling salesman problem and the flow shop scheduling problem [15,16]. Composite wing design is also one of the examples of permutation optimization [1]. In composite wing design, the main objective is to decide the layer stacking sequence by changing the order of layer angles for the given numbers of layers of each angle [1]. However, the standard crossover operator may lead to inadmissible design individuals, which do not maintain the numbers of plies of each orientation, in the permutation optimization problem. If the standard crossover operator is used, “gene repair” or “fixing-up” operation is required to handle the number-of-ply constraints. Furthermore, “gene repair” or “fixing-up” operation is also required in the permutation genetic algorithm if another design constraint is assigned, such as the contiguity condition [1,3]. To deal with these problems, a new permutation crossover operator was proposed, i.e., gene-rank GA in the composite wing design [1].

In this study, we employ a memory approach and a local improvement by the permutation operator to improve the numerical efficiency of the genetic algorithm for multidisciplinary laminate design. We classify the design criteria into two groups: layer combination dependent criteria and layer sequence dependent criteria. Then, the permutation operation is done for the identical design individual, stored in memory, to improve the performance for layer sequence dependent design criteria. In the permutation operation, we suggest “random shuffle” as a stochastic operator and “learning from experience” as an intentional local improvement operator.

The sample problem of weight minimization of laminated plates under two design constraints is treated to validate the numerical efficiency of the current approach. The numerical efficiencies of the proposed approaches are compared with that of the standard genetic algorithm with memory.

2. Design criteria and variables

In the laminate design, a number of structural design criteria are considered: for example, structural mass, stiffness, strength (failure resistance), buckling, and aero-elastic efficiencies [17,18,23,26]. If the liquid composite molding technique is selected as the manufacturing route, process requirements must also be taken into account, such as mold filling time, mold clamping force, and maximum pressure [19–22]. To meet design criteria, a number of design variables should be decided.

Fiber orientation and layer stacking sequence are usually the two main design variables in the laminate design of the prepreg layup. If the several pre-assigned layer angles (e.g., 0° , $\pm 45^\circ$, and 90°) are considered due to manufacturing complexities, the opti-

mization problem becomes an integer programming process to decide the layer stacking sequence. In genetic algorithms, the pre-selected layer angles are encoded into the natural numbers representing “gene” and combined to one another to construct a string representing “chromosome,” which is a layup configuration in the laminate design. In the standard genetic algorithm, the string length of each chromosome, i.e., the number of layers in the laminate, participating in the crossover operation should be identical. Hence, some researchers have introduced the “empty stack” approach to deal with the variation of the number of layers [5,22]. On the other hand, fiber volume fraction, one of the most important parameters in the composite design, is not considered as design variable, because it is fixed in prepreg techniques. In this case, the total thickness of laminate is proportional simply to the number of layers [5]. In general, it is desirable to increase fiber volume fraction to improve the specific stiffness/strength and to reduce the density of composites. Meanwhile, an excessively high fiber volume fraction leads to manufacturing difficulty. Fiber volume fraction is a continuous variable, while layer stacking sequence and number of layers are discrete variables. In some works, fiber volume fraction has been encoded into a binary string in the genetic algorithm [23]. In this discrete encoding of a continuous variable, the string becomes too long, and the size of design space increases accordingly. Another approach is to consider the fiber volume fraction as a dependent parameter on the layer stacking sequence by the projection into a feasible domain [22].

If the fiber volume fraction is held constant in the stacking sequence design of laminates, we can classify the design criteria, in terms of laminate layup configuration, into two groups: layer combination dependent criteria and layer sequence dependent criteria. Let us consider two different layer stacking sequences of the same number of layers:

Layup 1 [90° 0° 0° 90°]

Layup 2 [0° 90° 90° 0°]

Both layups consist of the same combination of layer angles (two layers of 0° and two layers of 90°), but their sequences are different. In general, the mechanics of laminates is described by laminated plate theory (see Fig. 1):

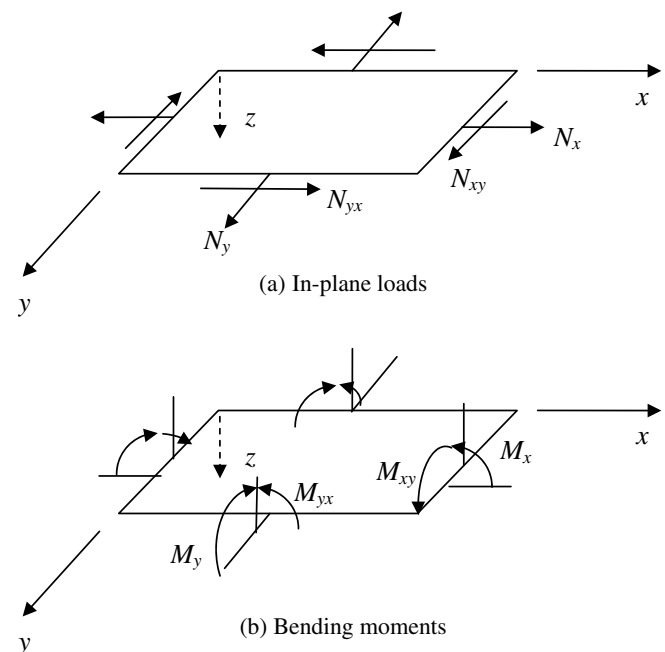


Fig. 1. Resultant loads on a laminated plate.

$$\begin{Bmatrix} \{N\} \\ \{M\} \end{Bmatrix} = \begin{bmatrix} [A] & [B] \\ [B] & [D] \end{bmatrix} \begin{Bmatrix} \{\epsilon^0\} \\ \{\kappa^0\} \end{Bmatrix} \quad (1)$$

$$([A], [B], [D]) = \sum_{k=1}^{N_L} \int_{h_{k-1}}^{h_k} (1, z, z^2) [Q]^k dz \quad (2)$$

where

$\{N\}, \{M\}$ resultant forces and moments, respectively;
 $\{\epsilon^0\}, \{\kappa^0\}$ mid-plane strain and curvature, respectively;
 $[A]$ extension stiffness matrix;
 $[B]$ extension-bending coupling stiffness matrix;
 $[D]$ bending stiffness matrix.

Both layups result in the same extension stiffness, A_{ij} , while they lead to a different bending stiffness, D_{ij} .

Effective planar mechanical properties of laminates can be obtained from the extension stiffness matrix, A_{ij} and the compliance matrix, a_{ij} :

$$\begin{bmatrix} A & B \\ B & D \end{bmatrix}^{-1} = \begin{bmatrix} a & b \\ b^T & d \end{bmatrix} \quad (3)$$

$$E_x = \frac{a_{11}}{H}, \quad E_y = \frac{a_{22}}{H}, \quad G_{xy} = \frac{a_{66}}{H} \quad (4)$$

In terms of in-plane stiffness, both layups lead to the same mechanical performance under planar loads. On the contrary, the two layups show different structural performances under the bending moments, if flexural stiffness or strength is considered as a design criterion. In this paper, in-plane stiffness/strength is classified as a “layer combination dependent criterion”, which leads to the same performance as long as the combinations of layer angles are identical even though the layer stacking sequences are different. Flexural stiffness/strength is an example of a “layer sequence dependent criterion” that results in different performances for different layer stacking sequences even though their layer combinations are identical. A classification of design criteria is presented in Table 1.

In terms of manufacturing, the mold filling time and the mold clamping force in the resin transfer molding process are “layer combination dependent criteria,” since the average permeability is identical for the same composition of layer angles:

$$K_{ij}^{\text{eff}} = \frac{1}{H} \sum_{k=1}^{N_L} h_k K_{ij,k} \quad (5)$$

$$H = \sum_{k=1}^{N_L} h_k \quad (6)$$

where h_k and $K_{ij,k}$ are the thickness and the permeability of each layer, respectively [19–22].

On the other hand, critical buckling load is a “layer sequence dependent criterion” when the bi-axial planar loads are applied to the symmetric laminates, since it depends on the bending stiffness matrix, D_{ij} :

$$\lambda_{mn} = - \frac{\pi^2 [D_{11}(m/a)^4 + 2(D_{12} + 2D_{66})(m/a)^2(n/b)^2 + D_{22}(n/b)^4]}{(m/a)^2 N_x + (n/b)^2 N_y} \quad (7)$$

where a and b are the dimensions of the laminate in coordinates x and y , respectively. The critical load factor λ corresponds to the combination of integer numbers m and n that minimizes λ_{mn} [19].

3. Memory

In the standard genetic algorithm, a new population may contain individuals that have already been evaluated in the previous generations. Furthermore, the same design individuals may also appear in the same generation. If the information on the same individuals, which have been already analyzed either in the previous generation or in the current generation, is saved in memory, a new fitness function evaluation is not needed and computing time can be reduced. If a binary tree algorithm is adopted to store the previous information, a gene-by-gene comparison is required for each design chromosome. Instead, we define identification numbers representing layer stacking sequence and layer angle combination.

Let us consider a laminate optimization problem with two design criteria: mold filling time as a layer combination dependent criterion and flexural stiffness as a layer sequence dependent criterion. In this optimization problem, design variables are layer stacking sequence and number of layers. Layer stacking sequence and number of layers are selected from the pre-assigned set:

$$\theta_k \in \{\alpha_1, \alpha_2, \dots, \alpha_n\} \quad (k = 1, N_L)$$

$$N_L \in \{N_{L,\min}, N_{L,\min} + 1, \dots, N_{L,\max}\}$$

Then, the layer stacking sequence of j th design individual is expressed as

$$[\theta]_j = [\theta_1, \theta_2, \dots, \theta_{N_L}]$$

We define an identification number for layer stacking sequence (ID_s) as

$$ID_s = \sum_{j=1}^{N_L} \sum_{k=1}^n k \cdot \delta(j-k) \cdot (n+1)^{j-1} \quad (8)$$

$$\delta(j-k) = \begin{cases} 1 & (\theta_j = \alpha_k) \\ 0 & (\theta_j \neq \alpha_k) \end{cases} \quad (9)$$

where N_L is the number of layers in the laminate and n is the number of angles in the pre-assigned angle set. $\delta(j-k)$ is zero for the “empty stack” that is used to match string length of different numbers of layers. The base “ $n+1$ ” is taken as the possible number of ply angles for the layer j , i.e., “ n ” for the number of ply angles in the pre-assigned angle set and “1” for the empty stack.

Then, an identification number for layer angle combination (ID_c) is defined as

$$ID_c = \sum_{j=1}^{N_L} \sum_{k=1}^n (N_{L,\max} + 1)^{k-1} \cdot \delta(j-k) = \sum_{i=1}^n n_i \cdot (N_{L,\max} + 1)^{i-1} \quad (10)$$

where n_i is the number of layers with angle α_i . The base “ $N_{L,\max} + 1$ ” is selected because the minimum number of the ply angle α_i in the laminate is “zero” and the maximum is “ $N_{L,\max}$ ”.

Consider the following case of four design individuals. For convenience, the population size is assumed to be 4, and the maximum number of layers is 12:

Table 1
Classification of design criteria in the laminate design

Layer combination dependent criteria	Layer sequence dependent criteria
In-plane properties (stiffness, conductivity)	Out-of-plane (flexural) stiffness/strength
Mold filling time	Interlaminar shear strength
Mold clamping force	Buckling
	Vibration (natural frequency)
	Aeroelastic efficiencies

(1) $[x_1 x_2 x_2 x_3 x_1 x_4 x_4 e e e e]$, ID _s = 380,311,	$\{2x_1 3x_2 1x_3 2x_4 4e\}$ ID _c = 4604
(2) $[x_1 x_2 x_3 x_4 x_1 x_2 x_3 x_4 e e e e]$, ID _s = 386,556,	$\{2x_1 2x_2 2x_3 2x_4 4e\}$ ID _c = 4760
(3) $[x_2 x_1 x_2 x_4 x_1 x_4 x_2 x_3 e e e e]$, ID _s = 279,307,	$\{2x_1 3x_2 1x_3 2x_4 4e\}$ ID _c = 4604
(4) $[x_1 x_2 x_2 x_3 x_1 x_4 x_4 e e e e]$, ID _s = 380,311,	$\{2x_1 3x_2 1x_3 2x_4 4e\}$ ID _c = 4604

The layer stacking sequence is expressed in $[\cdot]$, and the corresponding layer angle combination is expressed in $\{\cdot\}$. For the first design individual, identification numbers for stacking sequence and layer composition are calculated with layer stacking sequence expressed in $[\cdot]$ as

$$\begin{aligned} ID_S &= 1 \times 5^0 + 2 \times 5^1 + 2 \times 5^2 + 2 \times 5^3 + 3 \times 5^4 + 1 \times 5^5 \\ &\quad + 4 \times 5^6 + 4 \times 5^7 + 0 \times 5^8 + 0 \times 5^9 + 0 \times 5^{10} + 0 \times 5^{11} = 380,311 \\ ID_C &= 13^{1-1} + 13^{2-1} + 13^{2-1} + 13^{2-1} + 13^{3-1} + 13^{1-1} + 13^{4-1} + 13^{4-1} \\ &= 2 \times 13^{1-1} + 3 \times 13^{2-1} + 1 \times 13^{3-1} + 2 \times 13^{4-1} = 4604 \end{aligned}$$

In the standard genetic algorithm, four structural analyses (flexural stiffness) and four process analyses (mold filling time) are needed for the fitness function evaluations in the same generation if the population size is four. However, the individuals consisting of the same genes, called “clones”, appear in the population as the evolution process continues. In the above example, the fourth design individual is identical to the first one (the same ID_S and ID_C). Since the fourth design individual is the clone of the first individual, the fourth individual needs neither structural analysis nor process analysis. In addition, the third individual has the same layer composition (identical ID_C) as that of the first one, even though the layer stacking sequence is not identical (different ID_S). Thus, the mold filling time evaluation is not needed for the third individual. As a consequence, only three structural analyses (layer sequence dependent criterion) and two process analyses (layer combination dependent criterion) are needed.

However, a huge amount of computer memory is required if all the information on the individuals, through all the generations, is recorded. Previous investigations have shown that the memory approach is sufficiently efficient if the previous information is stored only for the latest generation [11]. In this study, we only store the information on the identical individuals appearing in the latest generation.

The present memory approach is summarized.

Algorithm 1. Memory approach for layer sequence (ID_S) and layer combination (ID_C)

Search for the given design individual in the memory by comparing ID_S and ID_C

```

If same  $ID_S$  (same layer stacking sequence) then
    Do not perform analysis for layer sequence dependent criterion
    Do not perform analysis for layer combination dependent criterion
Else
    Perform analysis for layer sequence dependent criterion
If same  $ID_C$  (same layer angle combination) then
    Do not perform analysis for layer combination dependent criterion
Else
    Perform analysis for layer combination dependent criterion
Endif
Endif

```

4. Local improvement by permutation

In this work, permutation is employed to improve the fitness of layer sequence dependent design criterion while maintaining the same performance for layer combination dependent design criterion. To construct a new stacking sequence of layers while maintaining the same layer angle composition, two strategies are adopted.

One is the “random shuffle”, i.e., the arbitrary rearrangement of the genes. Random numbers are generated and assigned to each gene in the chromosome. Then, the genes are rearranged in the increasing order of the assigned numbers. The other strategy is “learning from experience,” in which the genes are rearranged intentionally. In general, auxiliary information, such as sensitivity or gradient, is needed for the local improvement of fitness. Furthermore, the traits of the better design individual are not known a priori during the search. Thus, a simple strategy is introduced to acquire the information on the traits of better design individuals. A target string is assigned to show the best performance for layer sequence dependent criterion and an attempt is made for a current string to resemble a target string as much as possible. In fact, the best way to obtain a target string is to perform an independent structural optimization. However, this way requires an additional computational burden. Instead, the target string is selected from the previous individuals and updated as the global search continues. For each number of layers, the design individual that has shown the best performance for the layer sequence dependent criterion during the genetic search is selected and assigned as a target string. This method is based on the idea of learning by Michalski [24]:

“Learning is constructing or modifying representations of what is being experienced.”

In the present study, the information by learning from experience is accumulated through the generations, not confined to the lifetime of an individual. It is a concept that an individual in the population tries to mimic the best prototype that ever appeared in the population through the history.

To rearrange genes so that a new string may resemble a target string as much as possible, a genetic algorithm is applied. In this work, it is assumed that the fitness function evaluation needs more computational efforts than the genetic operations. The main purpose of gene arrangement is the enhancement of the layer sequence dependent criterion. Because the layer angle composition is preserved, the performance of layer combination dependent criterion remains unchanged. We can utilize the fact that the outer layer, generally, affects the layer sequence dependent criteria such as bending stiffness and buckling load, more than the inner layer does, since these criteria are influenced by the bending stiffness matrix D_{ij} in Eq. (2) of the laminate theory.

Consider the following case: Through genetic search, the target string has been found to show the best performance for the layer sequence dependent criterion among the previous design individuals whose number of layers is N_L . Then, a new string is found that has the identical layer stacking sequence as one of the design individuals stored in the memory. This clone string is not necessarily identical to the target string:

Target string: $[a_1 \ a_2 \dots a_{N_L}]$, N_L = number of layer;
Clone string: $[b_1 \ b_2 \dots b_{N_L}]$.

To rearrange the genes in the clone string, an objective function is formulated as

$$\text{Minimize} \quad \sum_{j=1}^{N_L} (a_j - b_j)^2 \times z_j^2 \quad (11)$$

where

a_j angle of j th layer in the target string;
 b_j angle of j th layer in the clone string (in which genes are to be permuted);
 z_j the distance from the center line of the laminate.

The weighting coefficient z_j^2 is selected, as many of the layer sequence dependent criteria are influenced by the bending stiffness D_{ij} in the laminate theory.

For example, it is assumed that layer angles are selected from the pre-assigned set of four orientations:

$$a_j, b_j \in \{0^\circ, 45^\circ, 90^\circ, -45^\circ\}$$

If the difference between the angle in the target string and in the clone string exceeds 90° , the angle difference is replaced by the difference between 180° and this value:

$$\text{If } |a_j - b_j| > 90^\circ, \text{ then } |a_j - b_j| \rightarrow 180^\circ - |a_j - b_j|$$

With the conventional genetic operators, i.e., crossover and mutation, the original layer angle composition may not be preserved. This gene arrangement problem has similarities with the traveling salesman problem to find an optimal route for visiting n cities and returning to the point of origin, where the inter-city distances are known [15]. Both problems aim at finding out an optimal permutation of given genes [15]. To deal with the problem in preserving the given combination of genes, a genetic algorithm is usually involved with a “fixing-up” procedure [3]. To cope with this problem in the traveling salesman problem, asexual reproduction, a simple and effective method without “fixing-up,” was suggested as an alternative to the standard crossover, by Chatterjee et al. [15].

Asexual reproduction is adopted to find optimal permutation of laminate layout in this work. In one parent (asexual) scheme, a child is produced by a parent. First, three cutting positions are selected at random, and the string is divided into four sub-strings:

$$[b_1 \ b_2 | b_3 \ b_4 \ b_5 | b_6 \ b_7 | b_8] = [A \ B \ C \ D]$$

$$[A] = [b_1 \ b_2], \quad [B] = [b_3 \ b_4 \ b_5], \quad [C] = [b_6 \ b_7], \quad [D] = [b_8]$$

Two other sub-strings of the reverse sequence are defined and denoted with the superscript $*$:

$$[B^*] = [b_5 \ b_4 \ b_3], \quad [C^*] = [b_7 \ b_6]$$

Then, we have eight possible ways for recombination of sub-strings:

- (1) $[A \ B \ C \ D] = [b_1 \ b_2 | b_3 \ b_4 \ b_5 | b_6 \ b_7 | b_8]$
- (2) $[A \ B \ C^* \ D] = [b_1 \ b_2 | b_3 \ b_4 \ b_5 | b_7 \ b_6 | b_8]$
- (3) $[A \ B^* \ C \ D] = [b_1 \ b_2 | b_5 \ b_4 \ b_3 | b_6 \ b_7 | b_8]$
- (4) $[A \ B^* \ C^* \ D] = [b_1 \ b_2 | b_5 \ b_4 \ b_3 | b_7 \ b_6 | b_8]$
- (5) $[A \ C \ B \ D] = [b_1 \ b_2 | b_6 \ b_7 | b_3 \ b_4 \ b_5 | b_8]$
- (6) $[A \ C \ B^* \ D] = [b_1 \ b_2 | b_6 \ b_7 | b_5 \ b_4 \ b_3 | b_8]$
- (7) $[A \ C^* \ B \ D] = [b_1 \ b_2 | b_7 \ b_6 | b_3 \ b_4 \ b_5 | b_8]$
- (8) $[A \ C^* \ B^* \ D] = [b_1 \ b_2 | b_7 \ b_6 | b_5 \ b_4 \ b_3 | b_8]$

In the present work, equal probabilities are given for choosing any arrangement.

As an alternative to mutation, “stack swap” is adopted [7]. In a “stack swap” two genes are arbitrarily selected and interchanged:

$$[b_1 \ b_2 \ \mathbf{b_3} \ b_4 \ b_5 \ b_6 \ \mathbf{b_7} \ b_8] \\ \rightarrow [b_1 \ b_2 \ \mathbf{b_7} \ b_4 \ b_5 \ b_6 \ \mathbf{b_3} \ b_8]$$

Asexual reproduction and stack swap are performed if a randomly-generated number exceeds a given probability value. The probabilities of asexual reproduction and stack swap are given as 0.8 and 0.2, respectively. The population size is assigned as 10. To construct a mating pool, a tournament selection scheme is used [25].

These strategies are presented in Algorithm 2. In this algorithm, the probability of random shuffle for permutation (P_p) is introduced. A random number (greater than 0.0 and smaller than 1.0) is generated and compared with the probability of random shuffle. If this generated number is exceeds the probability of random shuffle, the gene rearrangement is done arbitrarily (random shuffle). If not, the genes are rearranged intentionally by genetic algorithm with asexual reproduction (learning from experience).

If the probability of random shuffle is zero, all the permutation operations are submitted to the random shuffle. If the probability is one, all the permutation operations are submitted to the learning from experience:

Algorithm 2. Memory approach for layer sequence (ID_S) and layer combination (ID_C)

Search for the given design individual in the memory by comparing ID_S and ID_C

If same ID_S (same layer stacking sequence) **then**

Generate a random number, $R(0 < R < 1)$

If $R > P_p$ **then**

Perform “random shuffle”

Else

Perform “learning from experience (sub-optimization by GA)”

Endif

Perform analysis of new design for layer sequence dependent criterion

If performance for layer sequence dependent criterion is improved **then**

Replace the original design by the new design

Else

Keep the original design

Endif

Else

Perform analysis for layer sequence dependent criterion

If same ID_C (same layer angle combination) **then**

Do not perform analysis for layer combination dependent criterion

Else

Perform analysis for layer combination dependent criterion

Endif

Endif

5. Design optimization problem

The optimization problem for minimal mass of laminated plates is considered to validate numerical efficiencies of proposed schemes. Two design constraints are considered. For the layer sequence dependent criterion, we consider a structural constraint, which is a stiffness requirement that restricts the maximum strain norm of the structure under bending load to a specific value. The strain norms are calculated with planar principal strains (ε_1 and ε_2) at the bottom and the top surfaces, respectively. The greater value of the two is selected as the strain norm of the laminate:

$$\varepsilon = \max_{\text{top, bottom}} \left[\sqrt{0.5 \times (\varepsilon_1 + \varepsilon_2)^2 + 0.5 \times (\varepsilon_1 - \varepsilon_2)^2} \right] \\ = \max_{\text{top, bottom}} \left[\sqrt{\varepsilon_1^2 + \varepsilon_2^2} \right] \quad (12)$$

The smaller the strain norm ε is, the stiffer the composite laminate becomes.

As the layer combination dependent criterion, we consider a process constraint, which is mold filling time requirement, in the resin transfer molding process. It is presented how to obtain the mold filling time, in Ref. [22]. Planar dimensions of rectangular plates are fixed as 0.4 m and 0.2 m in x and y coordinates, respectively. A bending moment of 500 N (moment per unit length) is applied to the plate in the y direction. To achieve such purposes, four variables, namely, layer stacking sequence, number of layers, thickness of the plate and fiber volume fraction should be optimized:

$$\begin{aligned} &\text{Minimize } W(x_i) = W((\theta_j)_i, H_i, N_{L,i}, V_{f,i}) \\ &\text{subject to} \\ &\begin{cases} \varepsilon < \varepsilon_c : \text{stiffness requirement} \\ t < t_c : \text{mold filling time requirement} \end{cases} \end{aligned} \quad (13)$$

where

W weight of the structure;
 x_i design vector;
 $(\theta_j)_i$ layer stacking sequence;
 H_i thickness of final product;
 $N_{L,i}$ number of layers;
 $V_{f,i}$ fiber volume fraction;
 ε strain norm;
 t mold filling time;
 ε_c maximum allowable strain norm;
 t_c maximum allowable mold filling time.

From the investigation, the original problem can be simplified. The fiber volume fraction can be obtained from the number of layers and the thickness of part. In addition, the thickness is regarded as a parameter to be decided from the layer stacking sequence and the number of layers [20,22]:

$$V_{f,i} = V_{f,i}(H_i, N_i) \quad (14)$$

$$H_i = H_i((\theta_j)_i, N_i) \quad (15)$$

The original problem is redefined with two design variables, which are the number of layers and the layer stacking sequence, as shown in the equation:

$$\begin{aligned} &\text{Minimize } W(x_i) = W((\theta_j)_i, N_i) \\ &\text{subject to} \\ &\begin{cases} \varepsilon < \varepsilon_c : \text{stiffness requirement} \\ t < t_c : \text{mold filling time requirement} \end{cases} \end{aligned} \quad (16)$$

To solve the optimization problem, the genetic algorithm is employed. Layer angles are selected from the pre-assigned set of orientations $\{0^\circ, +45^\circ, 90^\circ, -45^\circ\}$. The minimum number of layers is eight and the maximum number of layers is 12. Population size is set to 40 in the genetic algorithm. To consider the variation of the number of layers, “empty stack” and “layer shift” are adopted for the crossover operation and “layer addition/deletion” is adopted for the mutation operation [5,22]. For the details on the genetic operations, see the previous work [22]. Constituent materials and their properties are listed in Tables 2 and 3.

If the standard genetic algorithm is applied, two structural analyses (for layer sequence dependent criterion) and one process analysis (for layer combination dependent criterion) are required to evaluate the fitness of each design individual.

Table 2
Mechanical properties of constituent materials

	Epoxy resin	Graphite fiber
Tensile modulus	2.6 GPa	228 GPa
Shear modulus	984.8 MPa	89.8 GPa
Poisson ratio	0.32	0.27

Table 3
Properties and processing condition for mold filling time estimation

Viscosity	0.2 Pa s
Longitudinal permeability	$1.0 \times 10^{-9} \text{ m}^2$
Transverse permeability	$1.33 \times 10^{-10} \text{ m}^2$
Injection pressure	0.1 MPa

Permeabilities are measured at the fiber volume fraction of 0.4.

6. Results and discussion

The numerical efficiencies of the proposed schemes are demonstrated by comparing the numbers of fitness function evaluations to the convergence. It is assumed that the fitness function evaluations require more computational costs than the genetic operations. It is trivial that the genetic algorithm with memory is more efficient than the standard genetic algorithm without memory. Hence, we employ the memory approach for all the cases. Then, we compare the numerical efficiencies of three different optimization strategies:

- I. memory and no permutation;
- II. memory and permutation (random shuffling, $P_p = 0.0$);
- III. memory and permutation (learning from experience, $P_p = 1.0$).

Before testing the above optimization strategies, 25 independent runs of the standard genetic algorithm were performed. In this case, a solution was assumed to be converged if there was no more improvement of fitness during the 200 generations. Then, the best solution of 25 runs was regarded as the global optimal solution.

Given the global optimum, the numerical efficiency was evaluated for each optimization strategy, by counting the number of fitness function evaluations until the convergence occurred. The genetic search was stopped when the global optimal solution (obtained a priori) was obtained. The numbers of process analyses (layer combination dependent criterion) and structural analyses (layer sequence dependent criterion) were counted for each numerical strategy. For each strategy, 25 independent runs were performed with different random initial populations.

For each numerical strategy, five optimization problems with different design constraints were solved. The design constraints considered and the corresponding solutions are presented in Table 4.

As the evolution proceeds, clone individuals with the same layer stacking sequence begin to appear in the population. If local improvement by permutation is not adopted, neither structural nor process analyses are performed for clone individuals. If permutation is applied, the genes of the clone individuals are rearranged and another structural analysis is needed to examine the improvement of performance for layer sequence dependent criterion (i.e., mechanical performance under out-of-plane loads). Thus, more computational efforts are required with local improvement by permutation if the same population size is used and the same generation passes by. Nevertheless for this additional computational effort, local improvement contributes to a faster convergence to a global optimum. As gene rearrangement methods, “random shuffle” and “learning from experience” were tested, respectively. The results of 25 independent runs for five design optimization problems are illustrated in Fig. 2. “RS” represents “random shuffle” and “LE” represents “learning from experience,” while “No permutation” represents a genetic algorithm without permutation while the memory approach is adopted.

Table 4
Optimal solutions for sample problems of laminate design optimization

Case	ε_c	t_c (s)	Weight (g)	Number of plies	Thickness (mm)	V_f	Stacking sequence
A	0.001	90	694.72	11	6.15	0.3574	90 ₅ 0 90 ₅
B	0.001	120	673.60	12	5.84	0.4112	90 ₄ 0 ₄ 90 ₄
C	0.001	150	656.56	12	5.66	0.4241	90 ₅ 0 ₂ 90 ₅
D	0.0008	120	749.02	13	6.52	0.3985	90 ₅ 0 90 0 90 ₅
E	0.0012	120	619.00	11	5.37	0.4100	90 ₄ 0 ₃ 90 ₄

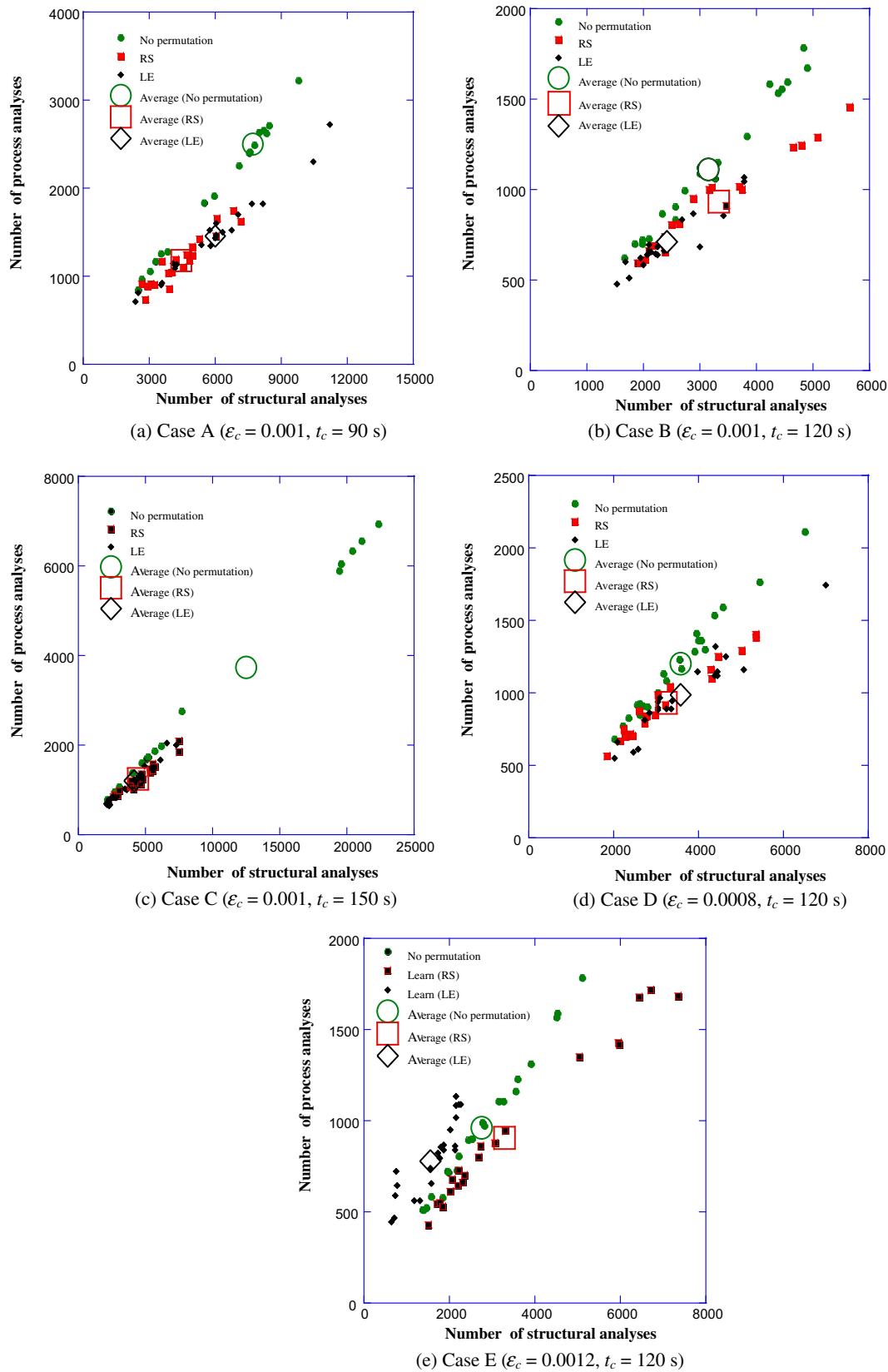


Fig. 2. Computational costs (the number of structural and process analyses).

The abscissa represents the number of structural analyses (for layer sequence dependent criterion) necessary to reach the global optimal solution, and the ordinate represents the number of pro-

cess analyses (for layer combination dependent criterion) required. The result of each run is plotted as a dot, and their average value is expressed as a large circle or rectangle. Local learning by permuta-

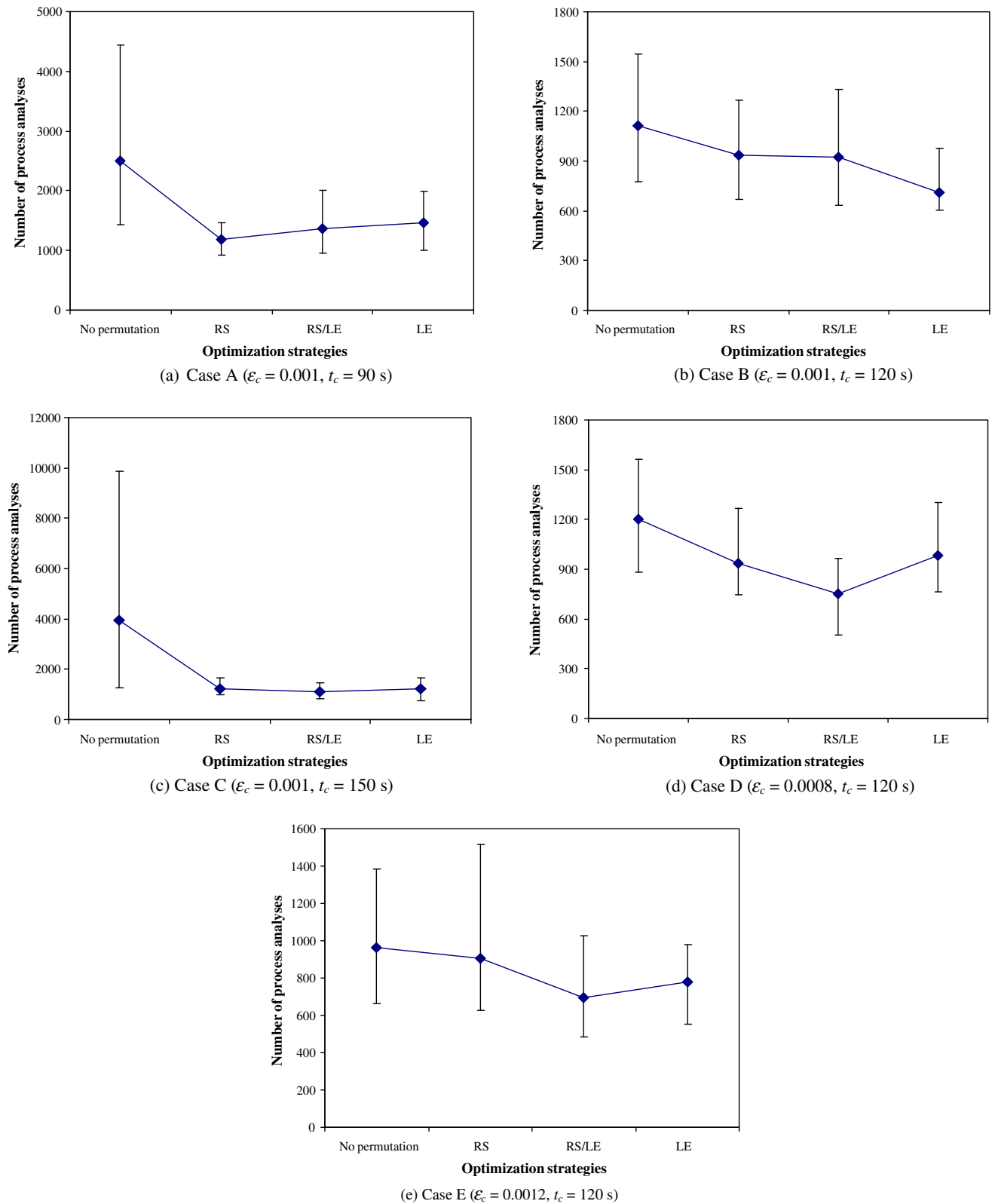


Fig. 3. Averages and standard deviations of the number of process analyses.

tion was able to reduce the computational efforts by as much as 70% (Case C). “Learning from experience” (GA with asexual reproduction) performed better than “random shuffle,” in most cases. However, “random shuffle” performed better than “learning from

experience” in Case A. In this case, “learning from experience” failed to improve bending stiffness (layer sequence dependent criterion), since the genetic search in the early generations didn’t find out a good prototype (target string) for a layer sequence dependent

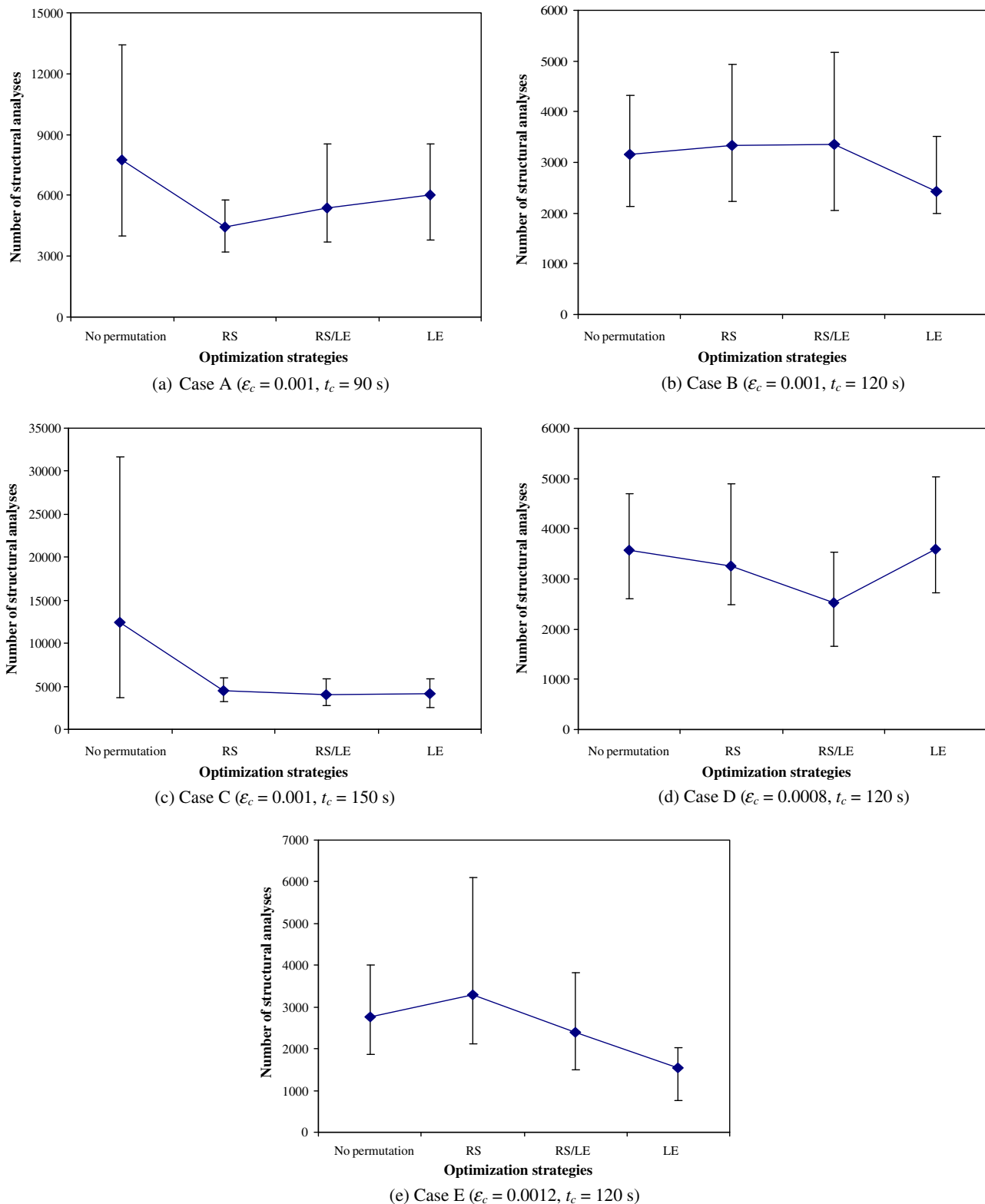


Fig. 4. Averages and standard deviations of the number of structural analyses.

criterion. Instead, “random shuffle” provides a jump from a local optimum to a global optimum by chance. Furthermore, “learning from experience” restricts the diversity in the population, since it rearranges the genes in a stereotyped manner. Hence, a combination of “random shuffle” and “learning from experience” may be a more reliable way. Subsequently, half of the clones were submit-

ted to “random shuffle” and the other half of the clones were submitted to “learning from experience” by setting the probability of random shuffle P_P to 0.5. This method is denoted as “RS/LE” in Figs. 3 and 4.

The averages and the standard deviations of the numbers of process and structural analyses to the convergence of 25 independent

runs are shown in Figs. 3 and 4, respectively. The upper and the lower standard deviations were computed, in each case. The lower error bar represents “average – lower standard deviation,” and the upper error bar represents “average + upper standard deviation”.

The numbers of process analyses for layer combination dependent criterion were reduced for all the strategies with permutation (Fig. 3). In fact, the number of analyses for layer combination criterion is identical for the same population size and the same number of generation, regardless of permutation adoption, since the layer combination does not change during the permutation operation. Hence, this improvement of numerical efficiency by permutation shows that the number of generations to the convergence was reduced. On the other hand, this fast convergence may be offset due to the additional structural analyses to examine the improvement of the performance for layer sequence dependent criterion, if the permutation is applied. Nevertheless, in most cases, the faster convergence by the local improvement of fitness by permutation surpassed the computational costs involved in additional fitness function evaluations.

We can see that “learning from experience” performed well on the whole. However, in Cases A and D, other optimization strategies (RS or RS/LE) showed better efficiencies than “learning from experience.” In Case A, “random shuffle” showed the best efficiency. The combination of “random shuffle” and “learning from experience” resulted in a good numerical performance in all cases.

7. Conclusions

New approaches for multidisciplinary optimization of composite laminates were proposed to save the number of fitness function evaluations in genetic algorithms. Design criteria in laminate design were classified into two groups, i.e., layer combination dependent criteria and layer sequence dependent criteria. Then, a memory approach was adopted to utilize the previous information on the same individuals appearing during the search. Permutation was performed for local improvement of the fitness of the same design individual. The genes were rearranged maintaining the same gene composition, and the fitness for layer sequence dependent criterion was compared with that of the original design individual. If the fitness was improved, the new individual replaced the original individual. Even though another fitness evaluation was required to examine the improvement of fitness by permutation, this strategy resulted in the reduction of the number of generations to the convergence and in the eventual improvement of numerical efficiency. Random shuffle and learning from experience (GA by asexual reproduction) were applied as permutation schemes. While random shuffle introduced a stochastic walk to escape from local optimum, learning from experience provided an intentional gene rearrangement to improve the performance of the design individual by imitating the best prototype that ever appeared during the evolutionary search. The proposed schemes were demonstrated to be numerically efficient in the multidisciplinary optimization of composite laminates.

References

- [1] Liu B, Haftka RT, Akgun MA, Todoroki A. Permutation genetic algorithm for stacking sequence design of composite laminates. *Comput Methods Appl Mech Eng* 2000;186:357–72.
- [2] Nagendra S, Jestin D, Gurdal Z, Haftka RT, Watson LT. Improved genetic algorithm for the design of stiffened composite panels. *Comput & Struct* 1996;58(3):543–55.
- [3] Todoroki A, Haftka RT. Stacking sequence optimization by a genetic algorithm with a new recessive gene like repair strategy. *Composites B* 1998;29:277–85.
- [4] Potgieter E, Stander N. The genetic algorithm applied to stiffness maximization of laminated plates: review and comparison. *Struct Opt* 1998;15:221–9.
- [5] Le Riche R, Haftka RT. Improved genetic algorithm for minimum thickness composite laminated design. *Composites Eng* 1995;5(2):143–62.
- [6] Soremekun G, Gurdal Z, Haftka RT, Watson LT. Composite laminate design optimization by genetic algorithm with generalized elitist selection. *Comput & Struct* 2001;79:131–43.
- [7] Le Riche R, Haftka RT. Optimization of laminate stacking sequence for buckling load maximization by genetic algorithm. *AIAA J* 1993;31(5):951–6.
- [8] Kogiso N, Watson LT, Gurdal Z, Haftka RT, Nagendra S. Design of composite laminates by a genetic algorithm with memory. *Mech Composite Mater Struct* 1994;1(1):95–117.
- [9] Gantovnik VB, Anderson-Cook CM, Gurdal Z, Watson LT. A genetic algorithm with memory for mixed discrete-continuous design optimization. *Comput & Struct* 2003;81:2003–9.
- [10] Kogiso N, Watson LT, Gurdal Z, Haftka RT. Genetic algorithms with local improvement for composite laminate design. *Struct Opt* 1994;7(4):207–18.
- [11] Povinelli RJ. Improving computational performance of genetic algorithms: a comparison of techniques. *GECCO 2000 late breaking papers*: p. 297–302.
- [12] Kim D, Choi YS, Lee SY. An accurate COG defuzzifier design using Lamarckian co-adaptation of learning and evolution. *Fuzzy Sets Syst* 2002;130:207–25.
- [13] Turner GW, Tedesco E, Harris KDM, Johnston RL, Kariuki BM. Implementation of Lamarckian concepts in a genetic algorithm for structure solution from powder diffraction data. *Chem Phys Lett* 2000;321:183–90.
- [14] Lamarck JB. Of the influence of the environment on the activities and habits of animals. *Zoological philosophy*. London: Macmillan; 1914.
- [15] Chatterjee S, Carrera C, Lynch LA. Genetic algorithms and traveling salesman problems. *Eur J Oper Res* 1996;93:490–510.
- [16] Liaw C-F. A hybrid genetic algorithm for the open shop scheduling problem. *Eur J Oper Res* 2000;124:28–42.
- [17] Saravanas DA, Chami CC. Multiobjective shape and material optimization of composite structures including damping. *AIAA J* 1992;30(3):805–13.
- [18] Watkins RI, Morris AJ. A multicriteria objective function optimization scheme for laminated composites for use in multilevel structural optimization schemes. *Comput Methods Appl Mech Eng* 1987;60:233–51.
- [19] Le Riche R, Saouab A, Bréard J. Coupled compression RTM and composite layup optimization. *Composite Sci Technol* 2003;63(15):2277–87.
- [20] Park CH, Lee WI, Han WS, Vautrin A. Weight minimization of composite laminated plates with multiple constraints. *Composite Sci Technol* 2003;63:1015–26.
- [21] Park CH, Lee WI, Han WS, Vautrin A. Simultaneous optimization of composite structures considering mechanical performance and manufacturing cost. *Composite Struct* 2004;65(1):117–27.
- [22] Park CH, Lee WI, Han WS, Vautrin A. Multi-constraint optimization of composite structures manufactured by resin transfer molding process. *J Composite Mater* 2005;39(4):347–74.
- [23] Pelletier JL, Vel SS. Multi-objective optimization of fiber reinforced composite laminates for strength, stiffness and minima mass. *Comput & Struct* 2006;84:2065–80.
- [24] Michalski RS. Understanding the nature of learning: issues and research directions. In: Michalski RS, Carbonell JG, Mitchell TM, editors. *Machine learning*, vol. II. San Mateo (CA): Morgan Kaufmann Publishers; 1986. p. 3–25 [chapter 1].
- [25] Goldberg D, Korb B, Deb K. Messy genetic algorithms: motivation, analysis, and first results. *Complex Syst* 1989;3:493–530.
- [26] Eschenauer HA, Schuhmacher G, Hartzheim W. Multidisciplinary design of composite aircraft structures by Lagrange. *Comput & Struct* 1992;44(4):877–93.