International Journal of Industrial Engineering and Operations Management (IJIEOM)
Volume. 1, No. 2, pp. 97-109, December 2019

IEOM Society International

International Journal of Industrial Engineering and Operations Management (IJIEOM)

Volume 1, No. 2, December 2019
pp. 97 - 110

Comparison Study of Discrete Optimization Problem Using Meta-Heuristic Approaches: A Case Study

Ali Ahmid, Thien-My Dao and Van Ngan LÊ

Department of Mechanical Engineering, ETS, Montreal, Canada
École de Technologie Supérieure
1100 Notre-Dame St W, Montreal, QC H3C 1K3

ali-elmbrok-salem.ahmid.1@ens.etsmtl.ca , thien-my.dao@etsmtl.ca , vanngan.le@etsmtl.ca

ABSTRACT

This paper presents the performance comparison of five meta-heuristic algorithms to solve a discrete optimization problem. The comparison is undertaken for a case of simply supported plate subjected to biaxial loading conditions. Furthermore, the optimization objective is to determine the optimal stacking sequence design of a laminate that maximizes the critical buckling load factor (λcb). The chosen meta-heuristics have been implemented using MATLAB with the same convergence criteria and the same maximum number of iterations to ensure a fair comparison. The implemented assessment criterion has performance measures of average CPU time, solution price, reliability, and normalized price. The results have demonstrated the outperformance of the Ant Colony Optimization Algorithm (ACOA) over other algorithms, which confirms the findings of previous studies. Moreover, the Tabu search algorithm (TS) and the Discrete Particle Swarm Optimization algorithm (DPSO) performed poorly due to their limited exploration capability. Additionally, the Genetic Algorithm (GA) and the Simulated Annealing algorithm (SA) exhibited a high level of reliability but showed an expensive solution cost. This study presents an adequate comparison approach of meta-heuristics, where it extends the comparison scope to cover the performance analysis of meta-heuristics more than that previously done in the domain of stacking sequence design optimization.

ARTICLE INFO

Received
March 5, 2019
Received in revised form
September 2019
Accepted
September 11, 2019

KEYWORDS
optimization, meta-heuristic, composite laminated plate, buckling load factor.

1. Introduction

The high competition in production design puts pressure on the designers to introduce a good quality and low-cost product that comply with the engineering standards. With limited resource context, the designer needs to benefit from all available resources and optimization techniques can solve such an issue. In early time, the gradient optimization techniques were commonly used in different engineering design applications due to their fast and accurate solutions. But with growing complexity and variety of applications, they became costly or incapable to find the Optima. On the other hand, metaheuristics exhibited a significant performance in solving optimization problems where the gradient methods failed to do so. The optimization of composite laminated structure design is an excellent example of such type of optimization problems (Kaveh, 2017).

Composite laminated structures are usually formed by laying several thin layers (plies) on top of each other and binding them with a matrix material. The combination of layers and matrix is called a laminate, which consists of microscale-

oriented fibers that emerge in the matrix material. The matrix material distributes and transforms the load over the fibers. Additionally, the tensile strength of the fibers is high in their orientated direction, whereas the matrix material has a high compression strength in any direction but has low tensile strength. The designer should select the right combination of both materials to achieve the optimal design. Several design variables should be appropriately determined such as a number of layers, the thickness of each layer, ply orientation angle, and stacking sequence that ensures the highest possible performance of the structure subjected to specific loading conditions (Vasiliev, 2017, Jones, 2014).

Buckling failure mode occurs suddenly when the composite laminated structure is exposed to compressive loading that exceeds a particular critical value. This failure mode is dangerous, especially for applications such as airplanes and ships, where human lives become threatened. The designer of composite laminated structures then must try to increase the capacity of the structure to bear the buckling load through optimizing the structure parameters. When the thickness of the structure is constant, the stacking sequence of the laminate turns into the significant design variable that can maximize the critical buckling loading of the structure (Nikbakt et al., 2018).

The optimization of the laminated composite structures is subjected to the design and manufacturing constraints such as a limited number of fiber orientations. The optimization of the laminate then becomes a hard-combinatorial optimization problem (Zein et al., 2016, Peeters and Abdalla, 2017, Rao, 2009). Ghiasi et al. (2010) reviewed different techniques used in the recent decades to optimize composite laminated designs and concluded that meta-heuristics are superior to gradient-based methods. Furthermore, (Nikbakt et al., 2018) reported the outperformance of meta-heuristics alongside gradient optimization algorithms due to their efficiency and stability. However, meta-heuristic algorithms are an ongoing optimization research domain to solve medium as well as large-scale problems that appear in different disciplines. Furthermore, trajectory-based meta-heuristics demonstrated a substantial local search capacity on its track to find the optimal solution, whereas population-based meta-heuristics exhibited a significant ability to explore the design space. Even though meta-heuristics, in general, could solve the discrete optimization problems efficiently, we still need to determine which algorithm outperforms the others for a specific problem according to the No Free Lunch theorem (NFL) by Wolpert and Macready (1997).

The literature is full of comparison studies of meta-heuristics that have been used to solve various engineering problems, e.g., TSP and scheduling, but little were devoted to the stacking sequence design problem. Furthermore, the previously published papers in the field were limited to two comparison approaches. First, the comparison of a newly developed algorithm (or enhanced version of a well-known algorithm) to previously published results of another meta-heuristic (Jing et al., 2015, Aymerich and Serra, 2008). Second, the selection of more than two algorithms and carrying out the performance comparison based on the author implementation of the meta-heuristics (Bloomfield et al., 2010). Both approaches brought valuable information that increased the knowledge about meta-heuristics performance as an optimizer of stacking sequence design. Although these comparison approaches are interesting, they still have some drawbacks such as the diversity of convergence criteria for the compared algorithms as in the first approach or the comparison limitation to one category of meta-heuristics as in the second approach.

To avoid this shortcoming, five different meta-heuristics were selected in this study to represent both population-based and trajectory-based meta-heuristics. The chosen algorithms frequently appeared in the literature of stacking sequence design optimization (Nikbakt et al., 2018). The five meta-heuristic algorithms have been implemented using MATLAB with the same convergence criteria and the same maximum number of runs to ensure a fair comparison. An assessment criterion has been performed by considering different performance measures such as average CPU time, reliability, and normalized price. Additionally, a well-known benchmarking problem was selected as a case study to carry out the comparison (Le Riche and Haftka, 1993, Kaveh et al., 2017). Eventually, the overall objective of this work is to develop an improved knowledge of optimization techniques and the selection of the most efficient algorithm that solves the stacking sequence design optimization problem.

2. Meta-heuristic Algorithms

Meta-heuristics are known as stochastic approaches that are frequently used in solving complex optimization problems. There are many classifications of meta-heuristics, and we have adopted the one illustrated in Figure 1, which classifies meta-heuristics into two categories population-based and trajectory-based. Moreover, five different algorithms were selected to represent both categories of meta-heuristics. Genetic algorithm (GA), Ant Colony Optimization (ACO), and Particle Swarms Optimization (PSO) are population-based meta-heuristics, whereas Simulated Annealing (SA) and Tabu Search (TS) are trajectory-based meta-heuristics. This section provides a short review of each meta-heuristic and the implementation structure of the algorithms used in this study.

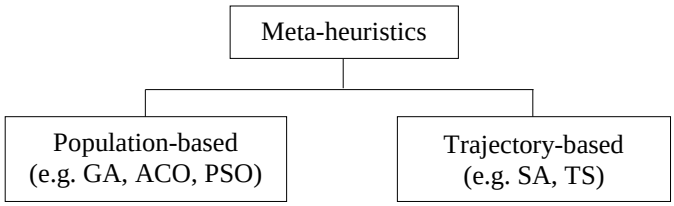


Figure 1: Metaheuristics classification

2.1. Genetic Algorithm (GA)

Holland suggested the original genetic algorithm in the 1960s, which was later detailed in its generally known form by Goldberg (1988). It is based on Darwin's theory of natural evolution, and it is implemented using elements of the natural genetics of reproduction, crossover, and mutation. GA shows its worthiness over classical optimization methods in solving composite laminated design optimization problems (Nikbakt et al., 2018). The significant adaptation of GA to optimize composite laminate design is credited to Le Riche (1993), as he proposed a modified GA that replaces binary coding of solution strings by integer coding. This formulation turned the binary GA algorithm into Permutation Genetic Algorithm (PGA). The results show a 2% reduction in the solution cost compared to binary GA(Le Riche and Haftka, 1993). The gene-rank GA introduced by Liu et al. (2000) is a permutation GA with gene-rank crossover operator. He compared his proposed GA with standard GA and older permutation GAs, and the gene-rank GA demonstrated better computational performance. Furthermore, Ehsani et al. (2016) used binary GA to determine the optimal stacking sequence of grid laminate by considering the different boundary conditions of the laminate edges. Moreover, GA algorithms are known for their expensive solution due to the slow convergence to the optimal solution. To overcome such drawback, Vosoughi et al. (2017) made hybrid GA with PSO algorithms as an operator to increase the convergence rate of standard GA. However, binary GA is still used as stacking sequence design optimizer. It offers a costly solution, while PGA demonstrates good performance for cheaper solutions.

In this study, the PGA structure was selected, as described by Le Riche (1993), to implement a GA program in MATLAB. Algorithm 1 illustrates the steps of PGA meta-heuristic. The algorithm is initialized by generating a random initial solution, and then it evaluates the fitness of the chromosomes (solutions). Based on their fitness value, the chromosomes are sorted from the maximum to the minimum, and the best-ranked individuals are selected for the reproduction process. To proceed with the reproduction, a pair of best individuals are randomly selected to be parents, and the crossover operator is applied to generate the children (new solution). Mutation and permutation operators are then applied to improve the new population exploration. This loop continues until the termination criteria is satisfied.

Algorithm 1: Permutation Genetic Algorithm procedure

Initialization:
Generate initial random population.
- Evaluate the population chromosomes fitness.

While (termination criteria not satisfied) **Do**
Select best-ranked individuals to reproduction.
Randomly select a pair of individuals to be parents
Apply crossover
Apply mutation to children
Apply permutation to children
Evaluate chromosomes fitness.
End PGA algorithm for discrete optimization problems

In the original PGA, which was proposed by Le Riche (1993), the different solutions have integer representation using 1, 2, and 3 numbers; where 1,2 and 3 represent $0^\circ, \pm 45^\circ$, and 90° fiber orientations, respectively. The laminate with $[\pm 45^\circ, 0^\circ{}_2, 90^\circ{}_2, \pm 45^\circ, \pm 45^\circ, 90^\circ{}_2, 0^\circ{}_2]_s$ stacking sequence have been represented by $[2\ 1\ 3\ 2\ 2\ 3\ 1]_s$. The different PGA operators of crossover, mutation, and permutation have been illustrated as follows:

Crossover: Parent #1: 3 2 **1 2 3 2** 2 1 → Child #1: 3 2 **1 3 2 1** 2 1
Parent #2: 2 2 **1 3 2 1** 3 2 → Child #2: 2 2 **1 2 3 2** 3 2

Mutation: Before: 3 2 1 2 3 **2** 2 1
After: 3 2 1 2 3 **3** 2 1

Permutation:

	1	2	3	4	5	6	7	8
Before:	2	2	**1**	**3**	**2**	**1**	3	2
After:	2	2	**1**	**2**	**3**	**1**	3	2

2.2. Ant Colony Optimization Algorithm (ACOA)

Dorigo (1991) developed the Ant Colony Optimization system that is inspired by the natural phenomena of the food searching strategy of the ant colony. He proposed a mathematical model that simulates this strategy of the cooperative attitude of an actual ant colony to find the optimal solution. He implemented his model to solve well-known optimization problems such as the travel salesman problem (TSP). The proposed model consists of four major steps. First, a suitable number of ants is assumed. Second, the probability of path selection is determined. Third, random numbers from 0 to 1 for each ant are generated. This step is repeated for all design variables, and it is followed by objective function evaluation and assessment. Fourth, the model checks the convergence process. ACO has been extended in different engineering areas to solve problems such as discrete structural design or composite laminated structures. Aymerich (2008) investigated the computational efficiency of ACO as an optimizer that maximizes the buckling load of a simply supported plate exposed to uniaxial loading. He compared the solution quality and robustness of ACO with GA and TS algorithms for the same reference case study, and the results show that the ACO algorithm has better performance. Furthermore, Koide et al. (2013) used the ACOA combined with finite element analysis to maximize the buckling load factor. They compared the obtained results of their proposed optimization solution with those previously obtained for GA by Le Riche (1993).

The structure of the proposed ACOA by Aymerich (2008) was mainly considered in this study, and Algorithm 2 summarizes the steps of the implemented ACO. The procedure starts with random initial laminate stacking being selected from the feasible solution set (possible fiber orientations). This step is followed by an evaluation of the objective function which will be stored in the ant routing table and used to generate a new feasible stacking sequence. Then, the local search movements of permutation and swap are applied to all generated solutions by ants in order to find better solutions. The local movement of permutation of ACO has the same effect of the permutation operator of PGA whereas the swap (also called two points mutation) movement occurs by randomly selecting and switching positions of two bits of the solution string. Finally, the global pheromone table is updated, according to Equation (1), where only the ants with the best solution deposited more pheromone trail on their path to the solution. This procedure continues until the termination criterion is satisfied.

Algorithm 2: Ant Colony Optimization Algorithm procedure

Initialization
Initialize the ACO parameters
While (termination criteria not satisfied) **Do**
  Construct Initial Solutions Table by Ants
  Evaluation
    Local Search:
Permutation
    • Swap
New Solutions evaluation
Apply Pheromones Updating Rule
End ACO algorithm for combinatorial optimization problems

The pheromone is updated according to the following rule:

$$\tau_{ij}^{k(t)} = (1-\rho).\tau_{ij}^{k(t-1)} + \sum_{k}^{n} \Delta\tau_{ij}^{k(t-1)} \quad (1)$$

where:

$$\Delta\tau_{ij}^{k(t-1)} = \sum_{k=1}^{m} \frac{f_k}{\sum_{k=1}^{m} f_k}$$

ρ denotes the evaporation rate, t is the current iteration number, m is the number of the optimal solution ants, and f_k is the fitness value of each ant.

2.3. Discrete Particle Swarm Optimization (DPSO)

Kennedy (1995) developed particle swarm optimization (PSO), and it is classified as a population-based metaheuristic. PSO is used to solve non-linear optimization problems with continues domain. PSO mimics the social behavior of a flock of birds, where each bird (called particle) moves with the flock according to two vectors of position and velocity. Each particle updates its position and velocity based on simple vector addition and subtraction until the optimal solution is found. Furthermore, PSO has various forms since it was invented, and the most popular one is known as G-best PSO. Additionally, PSO is known for its significant ability to explore a solution space with a fast convergence rate (Parsopoulos and Vrahatis, 2002). Different variants of PSO were developed to solve optimization problems of various engineering applications. Multiple versions of Discrete PSO (DPSO) were designed to solve combinatorial optimization problems such as stacking sequence optimization (Zadeh et al., 2018). Chang et al. (2010) proposed a new variant of DPSO called Permutation Discrete Particle Swarm (PDPSO). He used PDPSO to determine the optimum stacking sequence of a laminate subjected to the buckling load criteria.

DPSO algorithm, as described by Zadeh (2018), has been adopted in this comparison study. Algorithm 3 illustrates the different steps of implemented DPSO. It is initialized by selecting random swarm of particles and a set of possible solutions, and then this swarm fitness is evaluated. The best global position is devoted to the particle with maximum fitness in the initial swarm. The local particle speed and position update according to Equations 1 and 2 to generate a new swarm. The evaluation of the new swarm is then carried out, and the global best position is updated if the fitness of best local position of the new swarm is higher than the fitness of the stored best global position. This loop continues until the termination criterion is satisfied.

Algorithm 3: DPSO Algorithm procedure

Initialization:
Generate initial random swarm.
Evaluate the initial swarm speed and position.
While (termination criteria not satisfied) **Do**
Update swam speed and position
Evaluate the new swarm fitness
End DPSO algorithm for combinatorial optimization problems

The particle speed and positions are updated according to the following equations:

$$X_{k+1}^{i} = X_{k}^{i} + V_{k+1}^{i} \tag{2}$$

$$V_{k+1}^{i} = wV_{k}^{i} + c_1 r_1 \left(P_{best}^{i} - X_{k}^{i}\right) + c_2 r_2 \left(G_{best}^{g} - X_{k}^{i}\right) \tag{3}$$

where X_{k+1}^{i} and V_{k+1}^{i} represent the updated position and speed respectively

2.4. Simulated Annealing (SA)

In 1953, Metropolis presented the concept of the simulated annealing algorithm. It is based on the mathematical analogy of the thermal annealing process of critically heated metals. When the heated metal reaches the melting temperature, the molten molecules move randomly concerning each other. Continued reduction of the temperature limits the movement of these molecules and therefore, leads them to be highly ordered until the crystal state is reached, which represents the lowest internal strain energy. The cooling rate has a direct impact on achieving the crystal state; the faster rate will not provide the molecules enough time to form a crystal, and they will attain a polycrystalline state instead, which has higher strain energy. Therefore, the crystallization of molten metals needs a controlled rate of cooling to obtain the lowest strain energy state, and this process is called annealing (Kirkpatrick et al., 1983).

Since its introduction, SA has been used to solve several engineering optimization problems, including stacking sequence design. Lombardi et al. (1992) used SA to optimize the composite laminate buckling load for a plate subjected to biaxial loading under strain limits with iso-oriented and contiguous plies. They considered the range within 0.1% of the best solution as a near or optimal solution in the design space. Erdal et al. (2005) presented an improved version of SA called

Direct Simulated Annealing (DSA) to maximize the buckling load factor of the biaxially loaded laminate. DSA was developed by Ali et al. (2002) to handle continuous variable design problems based on memorizing the previous solutions and using a group of points instead of one point in its search for the optimal solution. Erdal (2005) adapted the DSA algorithm to optimize composite laminate design, which is a discrete optimization problem, and he investigated the performance of the algorithm by increasing the difficulty of the problem and increasing the design space size. He demonstrated that DSA performed well even with larger design space, and it overcame the cons of the original SA that was used by Lombardi(1992). Javidrad et al. (2017) proposed a modified SA algorithm that uses the parallelization concept, where the search is performed parallel to the multiple initial points, and the best-found solution is selected as the optimal solution. The convergence speed, for large design spaces, was a result of SA modification.

The structure of the standard SA algorithm was used to implement the algorithm in this comparison study, and the main steps of the implemented SA are listed in Algorithm 4. SA starts by generating an initial random solution and then computing the objective function value. The initial solution is considered as a current solution, and a new solution is randomly generated about it. The energy of the new solution is determined, which is also known as accepting probability, as shown in Equation 4. If the new solution energy is greater than the current solution energy, then the new solution becomes the current solution; otherwise, another new solution is generated. The temperature T is reduced if the SA loop iterations exceeds the certain number of iterations n. These actions are repeated until the termination criteria is satisfied (Rao and Rao, 2009).

Algorithm 4: Simulated Annealing Algorithm procedure
Initialization: Initialize SA parameters (T, c, n) Generate initial random solution. Evaluate the initial solution. **While** (termination criteria not satisfied) **Do** Generate a new solution from the current solution vicinity. Calculate the current solution energy. Calculate the new solution energy. Compare both solutions energy Update the current solution with the biggest. If the number of iterations > n Reduce the temperature by reduction factor c. **End** SA algorithm for combinatorial optimization problems

2.5 Tabu Search (TS)

Tabu Search (TS) is a local searching algorithm that explores the neighborhood of local optima. This algorithm uses a memory strategy to prevent recycling of old solutions. The original TS was presented by Glover (1991), and since then, it has improved and become widely used in solving combinatorial optimization problems such as TSP. Kaw et al. (2003) employed TS to optimize the stacking sequence of a rectangular laminate subjected to buckling loads. Three different loading cases have been investigated and compared to the previous results obtained using GA. The results illustrate a significant reduction in the solution cost by 25% and 55% for the first and second case, respectively, whereas a slight decrease of 1% was obtained in the third case. Kaw et al. (2003)concluded that TS is a competent optimization tool for stacking sequence problems, but it needs a favorable initial solution. Additionally, Rao (2007) used TS to enhance the local searching capability of the SA algorithm to optimize the stacking sequence and the new algorithm called TSA, which demonstrated superior performance to GA. Algorithm 5 explains the structure of the TS algorithm used in the current comparison work.

Algorithm 5: Tabu Search Algorithm procedure
Initialization: Generate initial random solution. Evaluate the initial solution. **While** (termination criteria not satisfied) **Do** Search the neighborhood: Permutation Swap Insertion Evaluate the new solutions Update the Tabu list. **End** TS algorithm for combinatorial optimization problems

TS algorithm is initialized by generating a random initial solution and then evaluating it. The neighborhood search of the initial solution is carried out by applying three different movements of permutation, swap, and insertion. The first two movements are similar to what have described in GA and ACO, where the insertion movement is imposed by selecting a random bit in the solution string and inserting it between two adjacent random bits. Next, the newly generated solutions are evaluated, and then the maximum fitness value updates the optimal solution. Then, the Tabu list is updated to prevent the next neighborhood search from returning to the previously selected solution. These steps of the algorithm continue until the termination criterion is satisfied.

3.1 Optimization Problem Statement

This assessment study focuses on the optimization of the stacking sequences that maximize the buckling load (strength) of the laminate as a case study. The objective function of maximizing the buckling load factor for laminate subjected to buckling load conditions could be written as follows:

$$max\,.\lambda_{cb}\left(p,q\right) \tag{1}$$

where λ_{cb} is the critical buckling load factor that buckles a simply supported plate subjected to in-plane loads of λN_X and λN_y into p and q half waves in x and, y directions. λ_{cb} could be defined, with respect to flexural stiffness, as:

$$\lambda_b(p,q)=\pi^2.\frac{\left[D_{11}\left(p/a\right)^4+2\left(\mathrm{D}_{12}+2\mathrm{D}_{66}\right)\left(p/a\right)^2+D_{22}\left(q/b\right)^4\right]}{\left(p/a\right)^2N_x+\left(q/b\right)^2N_y} \tag{2}$$

The smallest value of $\lambda_b(p,q)$ is considered as the critical buckling load factor**.** The critical values of p and q are linked to different factors such as laminate material, number of plies, loading conditions, and the plate aspect ratio. In uniaxial loading of a simply supported plate, the critical buckling load occurs when p=1 whereas in biaxial critical buckling loads, it needs to be determined as the minimum value of λ_{cb} (p,q) (Söyleyici, 2011).

3.2 Solution Representation and Design Space

The most commonly used fiber orientations are $0^\circ, \pm45^\circ$, and 90°. In meta-heuristic algorithms, the solution (stacking sequence) takes the form of a bit string that consists of a combination of plies with these angles. The different solutions are integrally coded with 1, 2, and 3 numbers, which respectively represent the three possible fiber orientations. For instance, the laminate with $[2\ 1\ 3\ 2\ 2\ 3\ 1]_s$ stacking sequence describes the laminate of $[\pm45^\circ, 0^\circ{}_2, 90^\circ{}_2, \pm45^\circ, \pm45^\circ, 90^\circ{}_2, 0^\circ{}_2]$s fiber orientations. The simplicity of using an integer representation and the significant performance, makes it the most widely used method in meta-heuristic optimization algorithms for composite laminated design (Le Riche and Haftka, 1995). The following formula could determine the Design Space Size (DSS) for a laminate represented by N plies:

$$DSS = K^N \tag{3}$$

where K is devoted to the number of ply orientation angles (e.g., K=3 for $0^\circ{}_2, \pm45^\circ, 90^\circ{}_2$)

3.3 Composite Laminate Design Constraints

The design of the composite should respect certain limitations of manufacturing and specific design considerations. In literature, some rules have been proposed to improve the effectiveness of a laminate design for different applications, (Zein et al., 2016, Peeters and Abdalla, 2017, Rao, 2009). The most used rules are classified and listed below:

- Manufacturing limitations: the thickness of the plies and fiber orientations are limited to the available manufactured values, which are usually integer, for ply thickness or certain angles such as ±45°, 0°, and 90° and for ply orientations. Additionally, the symmetrical laminate makes the manufacturing process more straightforward.

- Strength and stiffness considerations: the symmetry of laminate is necessary to prevent extension-bending coupling (i.e., $B_{i,j}$=0). Furthermore, the balanced laminate (which has pairs of plies with the same thickness and different signs of same orientation angle θ) condition is needed to avoid shear-extension coupling (i.e., A_{16}=A_{26}=0). All the plies with $\pm\theta$ will be grouped to minimize the effect of bending and twisting coupling. Moreover, the congestion of the same orientation plies should be limited to 4 plies for each group to develop a homogeneous laminate and reduce inter-laminate stresses and matrix crack failure. Furthermore, the stiffness degradation can be reduced by devoting 10% of the total number of plies for each orientation angle of 0°, ±45°, and 90°.

Generally, the constraints in stacking sequence optimization with constant laminate thickness t could be treated as:

- Symmetry constraint is enforced by optimizing half of the laminate.
- Balancing constraint is enforced by selecting θ_2 for the standard fiber orientation set of 0°, ± 45°, and 90°.
- Only N/4 ply orientations are needed to describe laminate because of balancing constraints.
- Contiguity constraint is handled by using the penalty parameter (p).

3.4 Objective Function Transformation

The handling of the constraints is the most critical aspect of the optimization problem formulation. The methods used with the algorithms reviewed here fall under one of the following categories:

- Feasibility-based rule.
- Discrete penalty functions.
- Hybrid approach.

A feasibility-based rule lets the algorithms generate the feasible candidate solutions only and then find the optimum one from them. Furthermore, the penalty functions are widely used in handling the constraints due to their simplicity with consistent results. Hybridization of both the previous methods could lead to an improvement in the performance of the algorithm to find the global optima (Le Riche and Haftka, 1993). More details about the constraints handling topic could be found in (Jiao et al., 2014, Barroso et al., 2017).

$$\lambda = (1-\mathrm{p}).max\,.\lambda_{cb}\left(p,q\right) \tag{4}$$

4. Comparison and Assessment Criteria

In addition to the elapsed time (average CPU time), literature has shown that other measures can be used to measure the computational effort of an algorithm. The first measure is price (P_S), which is defined as the number of objective function evaluations within a search run, and it reflects the computational cost of the search process. The second measure is practical reliability (PR) , and it is defined as the percentage of runs that achieve Practical Optima (PO) at a specific run. The last measure is the normalized price (nP_S) which is defined as the ratio of price and practical reliability. Practical optima PO is defined as the solution within 0.1% error value of the best possible solution (Malan and Engelbrecht, 2014, Le Riche and Haftka, 1993, Kogiso et al., 1994).

5. Benchmarking Numerical Example

MATLAB programs were written for each algorithm reviewed here, which are described in pseudo codes as illustrated in Algorithms 1-5. The literature has shown persistent development of new composite optimization solutions. To verify these new solutions, there is a crucial demand to select the well-known benchmarking problems. The widespread benchmarking problem used in the literature of stacking sequence optimization is accredited to Le Riche (1993), and it is indeed widely used in the reviewed studies of the current work. The original problem describes a simply supported plate subjected to an in-plane biaxial loading, as shown in Figure 2.

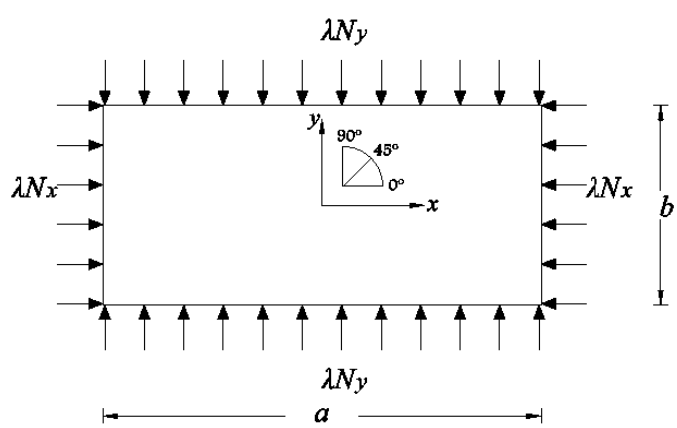


Figure 2: Simply supported plate subjected to biaxial loading.

The thickness of each ply t is assumed constant, and the plies' orientations are limited to 0°, ±45°, and 90° sets of angles. The number of plies N is constant. The required properties, dimensions, and loading conditions are listed in Tables 1 and 2. Furthermore, the objective function is maximizing the critical buckling load. The constraints are integrated to the solution (e.g., balanced laminate, symmetrical, etc.).

Table 1: Graphite-Epoxy lamina's properties (Koide et al., 2013)

Elastic Properties				Strength Properties				
E1 (GPa)	E2 (GPa)	G12 (GPa)	ν_{12}	X_T (MPa)	Y_T (MPa)	X_C (MPa)	Y_C (MPa)	S12 (MPa)
127.59	13.03	6.41	0.3	1500	40	1500	246	68

Table 2: Dimensions and loading conditions of composite laminated plate(Koide et al., 2013)

Dimensions				Loading	
# Plies NL	Thickness t (mm)	Length a(mm)	Width b (mm)	Nx (N/m)	Nx/Ny
64	0.127	508	25.4	175	1

The five implemented algorithms were tested on the same machine, as shown in Figure 3, for the same number of experiments; N_{exp}=200. This number was used in the original reference case by Le Riche (1993); he used it to tune the PGA parameters and at the same time to determine the performance of his proposed PGA.

```
 ========================================================================
  User: Ali Ahmid                                        15-May-2019 00:24:25
 ========================================================================
Machine Information:
                    CPU Processor: Intel(R) Core (TM) i7-4790 CPU @ 3.60GHz
                  CPU clock speed: 3601 MHz
             CPU Cache size (L2): 1024 KB
    Number of physical CPU cores: 4
 Installed physical memory (RAM): 16 GB
           Operating System Type: Windows
        Operating System Version: Microsoft Windows 7 Enterprise
 ========================================================================
```

Figure 3: The specifications of PC-machine used in the current comparison study.

An initial random solution was initialized for PGA, SA, and DPSO while TS started with the best solution of 10 random solutions. ACOA began with the initial pheromone of value .004. The number of practical optima was determined by considering the near-optimal solutions. In this research, the range of the practical optimal solutions was set to just 0.1% of the global optima.

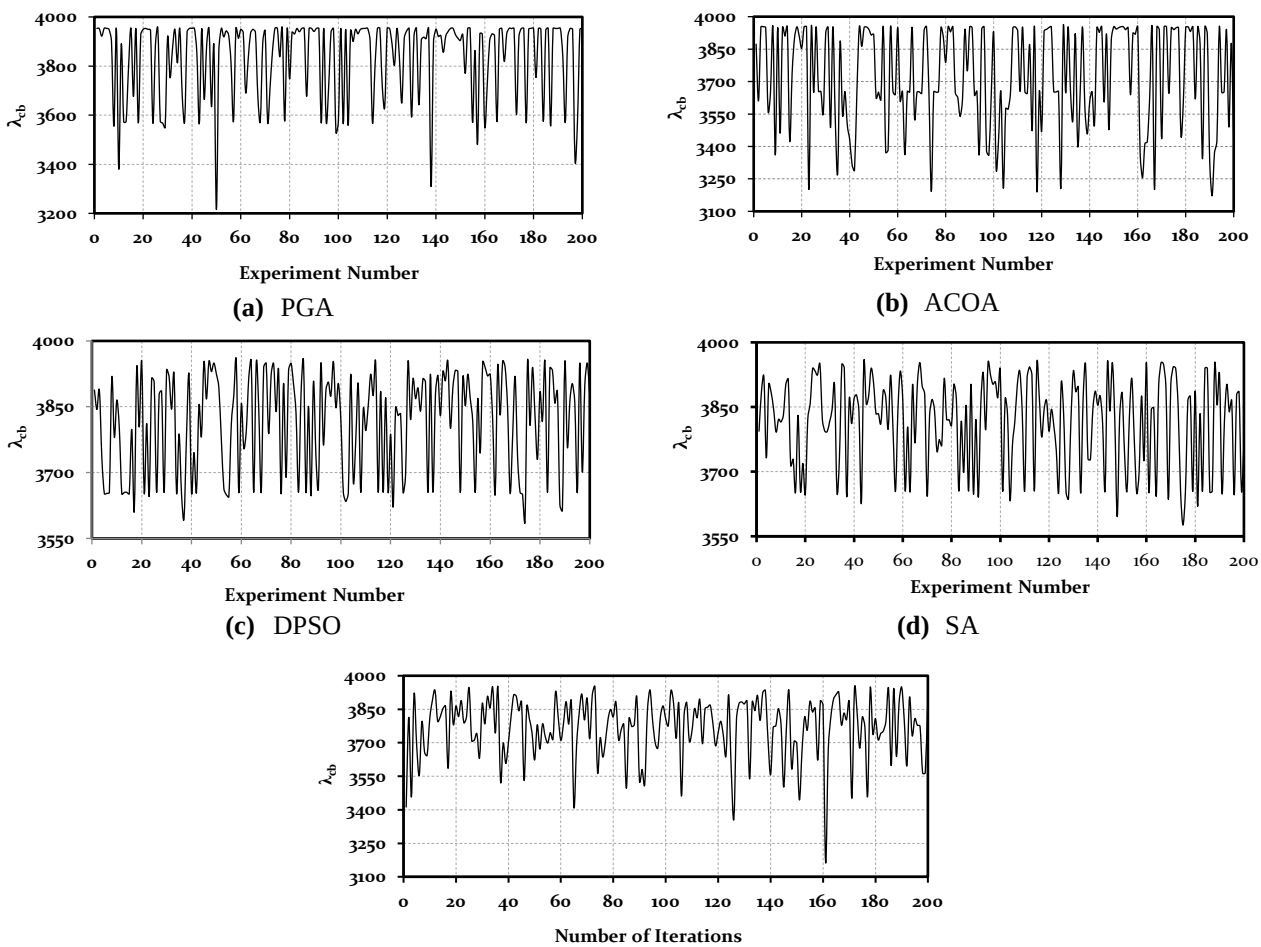
6. Results and Discussion

The obtained results for the biaxially loaded laminate are listed in Table 3. Additionally, the maximum critical buckling load λcb values were plotted versus their experiment number, as illustrated in Figure 4. Additionally, the convergence of each algorithm has been graphically illustrated in Figure 5. According to the introduced comparison and assessment criteria in section 4, different comparison measures of average CPU time, average price, reliability and normalized price were determined and have been illustrated in Figures 6-8.

The first four algorithms reached the same global optimal solution, whereas the Tabu Search algorithm missed it slightly, as presented in Table 3. The optimal stacking sequence followed the same pattern of switching between two groups of 902 and ±452 fiber orientations, which confirms the results of Erdal (2005), Aymerich (2008), and a more recent study by Kaveh (2017); however, 02 angle orientations did not exist in the global optimal solution. In Figure 5, the convergency of trajectory-based meta-heuristics form a series of steps line graph on its way to the optimal solution zone in the design space. On the other hand, the population-based meta-heuristics form a progressive curve graph to converge to the optimal solution. The numerical experiments confirm the random performance fluctuation of the meta-heuristic algorithms due to their stochasticity, as illustrated in Figure 4.

Table 3: The optimal stacking sequence for 64 ply laminates subjected to biaxial loading without contiguity constraint (Ny/Nx=1 and a/b=2).

Algorithm	Optimal Stacking Sequence	Critical Buckling Load Factor
PGA	$[90_2\ 90_2\ 90_2\ 90_2\ 90_2\ \pm45_2\ \pm45_2\ 90_2\ \pm45_2\ \pm45_2\ \pm45_2\ 90_2\ \pm45_2\ \pm45_2\ \pm45_2\ \pm45_2]$	3973.01
ACOA	$[90_2\ 90_2\ 90_2\ 90_2\ \pm45_2\ 90_2\ \pm45_2\ 90_2\ \pm45_2\ 90_2\ \pm45_2\ \pm45_2\ \pm45_2\ \pm45_2\ \pm45_2\ \pm45_2]$	3973.01
SA	$[90_2\ 90_2\ 90_2\ 90_2\ 90_2\ \pm45_2\ 90_2\ \pm45_2\ \pm45_2\ \pm45_2\ \pm45_2\ \pm45_2\ \pm45_2\ \pm45_2\ 90_2\ \pm45_2]$	3973.01
DPSO	$[90_2\ 90_2\ \pm45_2\ \pm45_2\ 90_2\ 90_2\ 90_2\ 90_2\ 90_2\ 90_2\ 90_2\ 90_2\ \pm45_2\ 90_2\ 90_2\ 90_2]$	3973.01
TS	$[90_2\ 90_2\ 90_2\ 90_2\ 90_2\ \pm45_2\ \pm45_2\ \pm45_2\ 90_2\ \pm45_2\ 90_2\ \pm45_2\ \pm45_2\ 90_2\ \pm45_2\ \pm45_2]$	3972.50



(a) PGA **(b)** ACOA **(c)** DPSO **(d)** SA (e) TS

Figure 4: Maximum Critical Buckling Load Factor vs. Experiment Number for the Five Meta-heuristics

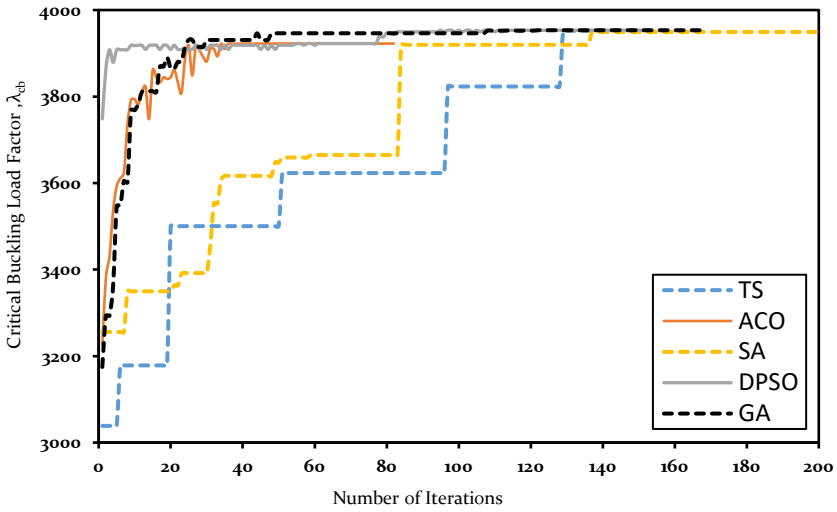


Figure 5: Meta-heuristics Convergence for Maximizing Critical Buckling Load Factor

In terms of computational effort, SA consumed the less CPU time, with just 2.75 sec to complete one run of the algorithm in average, and ACO became the second with 4 sec, while DPSO needed around 32 sec, as shown in Figure 6. The reliability values are shown in Figure 7(b) that demonstrates the outperformance of PGA and SA over ACOA, DPSO, and TS. However, in terms of the solution cost, ACOA ranks above all others, as it only costs 87.73 runs on average to reach the global optima with 76.5% reliability. Even though PGA and SA exhibit high reliability, they produced expensive solutions compared to ACOA, as shown in Figure 8.

However, the current work is devoted to providing a general overview of the performance of meta-heuristic algorithms. The critical strength factor that has been considered here is the buckling load factor only. Adding the strain failure factor to the optimization criteria could affect the final optimal stacking sequence design. Furthermore, the size of the design space is another factor that could have a significant impact as well (Todoroki and Haftka, 1998).

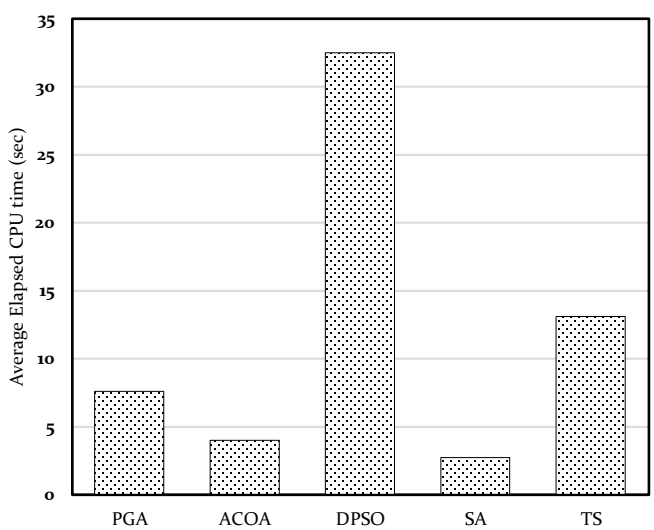


Figure 6: Average elapsed CPU time for the implemented meta-heuristic Algorithms

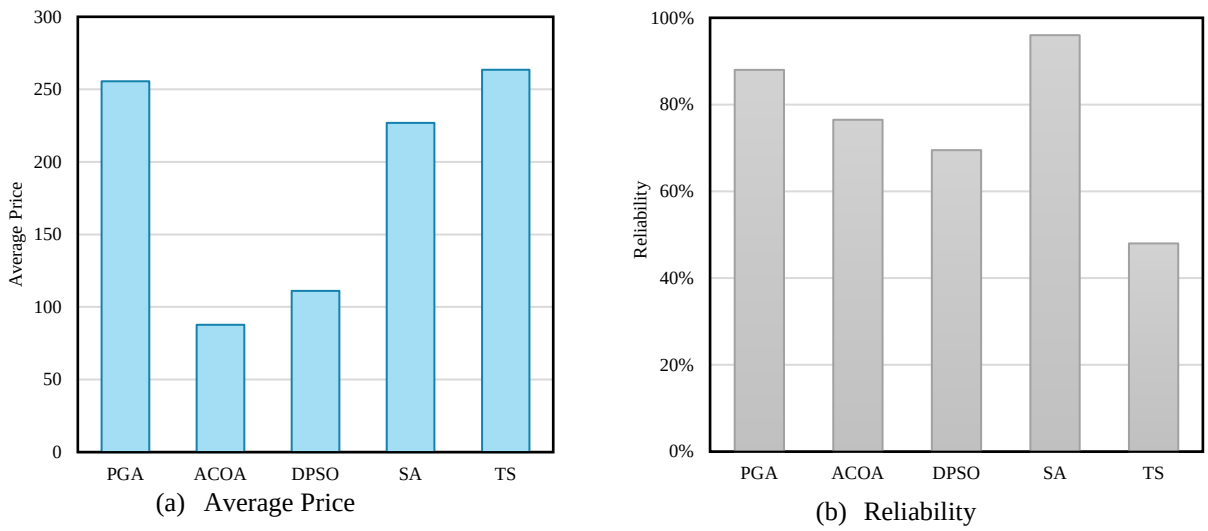


Figure 7: Average price and reliability for implemented meta-heuristic Algorithms

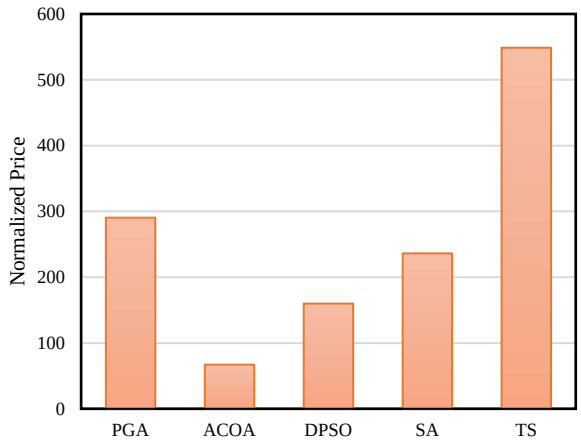


Figure 8: Normalized price comparison for Meta-heuristic

7. Conclusion

A comparison of meta-heuristic optimization techniques has been conducted in this article, and the basic knowledge of stacking sequence optimization fundamentals has been introduced. Five different well-known optimization algorithms have been implemented and examined. This research tried to bridge the gap of previous investigations such as comparing the performance of algorithms from the same category of meta-heuristics. In addition, it applied the same convergence criteria of the investigated meta-heuristics to ensure a fair performance assessment. It may be useful to note that the different meta-heuristics parameters that have been used here were taken from previous studies, and any parameters refinement is out of the current study scope.

The reliability analysis results reveal that GA and SA offer a more reliable solution than ACOA. In terms of solution cost, ACOA ranks above all others, as it only costs 87.73 runs on average to reach the global optima with 76.5% reliability, whereas DPSO, the nearest other meta-heuristic, costs 120 runs on average with 70.15 % reliability. Based on the current case study results, we can conclude that ACOA is a promising algorithm, and this agrees with previous studies of Bloomfield (2010) and Aymerich (2008). The significant performance of ACOA is expected, where it is basically designed to solve discrete optimization problems (Bloomfield et al., 2010).

ACOA could be improved by integrating other local search movements rather than only relying on permutation and swap movements (Dorigo and Stützle, 2019). GA has low local search performance, which could be improved by

combining it with other efficient local search algorithms such as PSO (Bloomfield et al., 2010). Eventually, further investigations are needed to verify this significant performance of ACOA, such as extend the comparison to include bigger design space or more design constraints with respect to stacking sequence design.

References

Ali, M. M., Törn, A. & Viitanen, S. 2002. A direct search variant of the simulated annealing algorithm for optimization involving continuous variables. *Computers & Operations Research,* 29**,** 87-102.

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

Barroso, E. S., Parente, E. & De Melo, A. M. C. 2017. A hybrid PSO-GA algorithm for optimization of laminated composites. *Structural and Multidisciplinary Optimization,* 55**,** 2111-2130.

Bloomfield, M. W., Herencia, J. E. & Weaver, P. M. 2010. Analysis and benchmarking of meta-heuristic techniques for lay-up optimization. *Computers & Structures,* 88**,** 272-282.

Dorigo, M. 1991. Ant Colony Optimization—new optimization techniques in engineering. *by Onwubolu, GC, and BV Babu, Springer-Verlag Berlin Heidelberg***,** 101-117.

Dorigo, M. & Stützle, T. 2019. Ant Colony Optimization: Overview and Recent Advances. *In:* GENDREAU, M. & POTVIN, J.-Y. (eds.) *Handbook of Metaheuristics.* Cham: Springer International Publishing.

Ehsani, A. & Rezaeepazhand, J. 2016. Stacking sequence optimization of laminated composite grid plates for maximum buckling load using genetic algorithm. *International Journal of Mechanical Sciences,* 119**,** 97-106.

Erdal, O. & Sonmez, F. O. 2005. Optimum design of composite laminates for maximum buckling load capacity using simulated annealing. *Composite Structures,* 71**,** 45-52.

Goldberg, D. E. & Holland, J. H. 1988. Genetic algorithms and machine learning. *Machine learning,* 3**,** 95-99.

Gürdal, Z., Haftka, R. T. & Hajela, P. 1999. *Design and optimization of laminated composite materials,* John Wiley & Sons.

Javidrad, F., Nazari, M. & Javidrad, H. 2017. Optimum stacking sequence design of laminates using a hybrid PSO-SA method. *Composite Structures*.

Jiao, L., Luo, J., Shang, R. & Liu, F. 2014. A modified objective function method with feasible-guiding strategy to solve constrained multi-objective optimization problems. *Applied Soft Computing,* 14**,** 363-380.

Jing, Z., Fan, X. & Sun, Q. 2015. Stacking sequence optimization of composite laminates for maximum buckling load using permutation search algorithm. *Composite Structures,* 121**,** 225-236.

Jones, R. M. 2014. *Mechanics of composite materials,* CRC press.

Kaveh, A. 2017. *Applications of metaheuristic optimization algorithms in civil engineering,* Springer.

Kaveh, A., Dadras, A. & Malek, N. G. 2017. Buckling load of laminated composite plates using three variants of the biogeography-based optimization algorithm. *Acta Mechanica,* 229**,** 1551-1566.

Kennedy, J. & Eberhart, R. Particle swarm optimization (PSO). Proc. IEEE International Conference on Neural Networks, Perth, Australia, 1995. 1942-1948.

Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. 1983. Optimization by simulated annealing. *science,* 220**,** 671-680.

Kogiso, N., Watson, L. T., Gürdal, Z. & Haftka, R. T. 1994. Genetic algorithms with local improvement for composite laminate design. *Structural optimization,* 7**,** 207-218.

Koide, R. M., De Franca, G. V. & Luersen, M. A. 2013. An ant colony algorithm applied to lay-up optimization of laminated composite plates. *Latin American Journal of Solids and Structures,* 10**,** 491-504.

Koide, R. M. & Luersen, M. A. 2013. Maximization of Fundamental Frequency of Laminated Composite Cylindrical Shells by Ant Colony Algorithm. *Journal of Aerospace Technology and Management,* 5.

Laguna, M., Barnes, J. W. & Glover, F. W. 1991. Tabu search methods for a single machine scheduling problem. *Journal of Intelligent Manufacturing,* 2**,** 63-73.

Le Riche, R. & Haftka, R. 1995. Improved genetic algorithm for minimum thickness composite laminate design. *Composites Engineering,* 5**,** 143-161.

Le Riche, R. & Haftka, R. T. 1993. Optimization of laminate stacking sequence for buckling load maximization by genetic algorithm. *AIAA journal,* 31**,** 951-956.

Lombardi, M., Haftka, R. & Cinquini, C. Optimization of composite plates for buckling by simulated annealing. 33rd Structures, Structural Dynamics and Materials Conference, 1992. 2313.

Malan, K. M. & Engelbrecht, A. P. 2014. Fitness landscape analysis for metaheuristic performance prediction. *Recent advances in the theory and application of fitness landscapes.* Springer.

Nikbakt, S., Kamarian, S. & Shakeri, M. 2018. A review on optimization of composite structures Part I: Laminated composites. *Composite Structures,* 195**,** 158-185.

Pai, N., Kaw, A. & Weng, M. 2003. Optimization of laminate stacking sequence for failure load maximization using Tabu search. *Composites Part B: Engineering,* 34**,** 405-413.

Parsopoulos, K. E. & Vrahatis, M. N. 2002. Recent approaches to global optimization problems through particle swarm optimization. *Natural computing,* 1**,** 235-306.

Peeters, D. & Abdalla, M. 2017. Design Guidelines in Nonconventional Composite Laminate Optimization. *Journal of Aircraft,* 54**,** 1454-1464.

Rao, A. R. M. 2009. Lay-up sequence design of laminate composite plates and a cylindrical skirt using ant colony optimization. *Proceedings of the Institution of Mechanical Engineers Part G-Journal of Aerospace Engineering,* 223**,** 1-18.

Rao, A. R. M. & Arvind, N. 2007. Optimal stacking sequence design of laminate composite structures using tabu embedded simulated annealing. *Structural Engineering and Mechanics,* 25**,** 239-268.

Rao, S. S. & Rao, S. S. 2009. *Engineering optimization: theory and practice,* John Wiley & Sons.

Söyleyici, M. U. 2011. *Stacking sequences optimization of the anti-buckled laminated composites considering various failure criteria.* İzmir Institute of Technology.

Todoroki, A. & Haftka, R. T. 1998. Stacking sequence optimization by a genetic algorithm with a new recessive gene like repair strategy. *Composites Part B: Engineering,* 29**,** 277-285.

Vasiliev, V. V. 2017. *Mechanics of composite structures,* CRC Press.

Wolpert, D. H. & Macready, W. G. 1997. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation,* 1**,** 67-82.

Zadeh, P. M., Fakoor, M. & Mohagheghi, M. 2018. Bi-level optimization of laminated composite structures using particle swarm optimization algorithm. *Journal of Mechanical Science and Technology,* 32**,** 1643-1652.

Zein, S., Madhavan, V., Dumas, D., Ravier, L. & Yague, I. 2016. From stacking sequences to ply layouts: An algorithm to design manufacturable composite structures. *Composite Structures,* 141**,** 32-38.

Biographies

Professor Thien-My Dao's areas of interest include the design, implementation, and computer-aided management of manufacturing systems. Among other things, his research and development activities focus on applied research in the field of performance and reliability analysis, the optimal design of manufacturing systems through simulation, the approach of neural systems, fuzzy logic and by the concept of "cell production / just in time". Professor Dao has worked for four years in industry. He is a member of Quebec Engineers Association since 1975, Canada Welding Institute (CAI) since 1985 and The American Society of Mechanical Engineers (ASME) since 1990.

Professor Van Ngan Lê's areas of interest include the product design, structural analysis, computer-aided engineering (CAE), computer aided design (CAD) and Finite Elements Methods (FEM). He teaches engineering courses related to his areas of interest. Prof. Lê worked five years for industry for Canadian companies like Combustion Engineering of Canada Ltd and United Air Craft of Canada as designer and stress analysis expert. He is also member of different organizations such as Quebec Engineers Association, Canada Welding Bureau (CWB) and American Society of Mechanical Engineers (ASME).

Ali Ahmid is a PhD. student at École de Technologie Supérieure (ÉTS) – Université du Québec. He holds a Bachelor of Science degree in Mechanical Engineering from Tripoli University –Libya. He has also earned his Master of Science degree in Mechanical Engineering from Cranfield University –United Kingdom. Ali has worked for research sector for more than ten years in field of product design and reverse engineering. His PhD. research is related to optimize and improve of composite materials products and structures using Computer Aided Engineering (CAE) and Meta-heuristic optimization algorithms.