 doi:10.20944/preprints202108.0573.v1

Article

ENHANCED HYPER-CUBE FRAMEWORK ACO FOR COMBINATORIAL OPTIMIZATION PROBLEMS

Ali Ahmid [1,] *, Thien-My Dao [2] and Ngan Van Le [3]

[1] Researcher;
[2] Mechanical engineering department, École de Technologie supérieure ÉTS; thien-my.dao@etsmtl.ca
[3] Mechanical engineering department, École de Technologie supérieure ÉTS; vanngan.le@etsmtl.ca
* Correspondence: ali-elmbrok-salem.ahmid.1@ens.etsmtl.ca

Abstract: Many combinatorial optimization problems are hard to solve within the polynomial computational time or NP-hard problems. Therefore, developing new optimization techniques or improving existing ones still grab attention. This paper presents an improved variant of the Ant Colony Optimization meta-heuristic called Enhanced Hyper Cube Framework ACO (EHCFACO). This variant has an enhanced exploitation feature that works through two added local search movements of insertion and bit flip. In order to examine the performance of the improved meta-heuristic, a well-known structural optimization problem of laminate Stacking Sequence Design (SSD) for maximizing critical buckling load has been used. Furthermore, five different ACO variants were concisely presented and implemented to solve the same optimization problem. The performance assessment results reveal that EHCFACO outperforms the other ACO variants and produces a cost-effective solution with considerable quality.

Keywords: Combinatorial optimization; Ant Colony Optimization (ACO); Buckling load factor; Composite laminate.

1. Introduction

Combinatorial optimization is devoted to the mathematical process of searching for optimal solution (maxima or minima) of an objective function with a discrete domain of decision variables. The possible number of solutions for a combinatorial optimization problem is equal to $[D]^n$, where D is the discrete design domain vector and n represents the number of design variables [1]. Therefore, the optimization problem becomes more computationally difficult to be solved when the number of design variables increases. Accordingly, many combinatorial optimization problems are hard to solve within deterministic polynomial time (or NP-hard). A Travelling Salesman (or TSP) is a typical example of this type of optimization problem where the number of cities to be visited is given and the shortest path is needed to be determined [2]. As the number of cities increases, the number of possible solutions increases too and this leads to the computational complexity of the problem, where it is not possible to enumerate all these solution possibilities with the limited computation resources, such as memory size or processor speed. Hence, to solve such problems, many optimization techniques have been developed.

The Ant Colony Optimization (ACO) algorithm demonstrated a significant performance improvement in solving NP-hard combinatorial optimization problems. The Traveling Salesman Problem (TSP) is a good example of such problems and it is solved using an early version of ACO [3]. The improvements in subsequent ACO algorithms focused on enhancing the algorithm variants to yield better searching and computational performance. As a result of the improved algorithm performance, many new applications of

 doi:10.20944/preprints202108.0573.v1

ACO appeared, such as the probabilistic TSP using estimation based ACO in Weiler, Biesinger [4] work. Gambardella and Dorigo [5] used a hybrid ACO with a novel local search tool to solve the Sequential Ordering Problem (SOP). Zheng, Zecchin [6] introduced a novel approach of ACO to optimize a water distribution system design. Furthermore, the optimization of the space truss design, using improved ACO, is demonstrated by Kaveh and Talatahari [7].

In the field of Stacking Sequence Design (SSD) of composite laminate, ACO has been used to determine the optimal stacking sequence design for maximizing the critical buckling load factor or natural frequency [8]. Aymerich and Serra [9] examined the ACO performance in solving stacking sequence design problem and he compared a modified standard ACO performance with Genetic Algorithm (GA) and Tabu Search (TS).He found that ACO performed much better than GA and TS in terms of solution cost and quality. Rama Mohan Rao [10] presented a hybrid ACO-TS algorithm to optimize the stacking sequence of a composite laminate subjected to bidirectional compression loading. He concluded that ACO is an effective optimization technique if combined with an appropriate local searching tool. Bloomfield, Herencia [11] conducted a comparison study of three meta-heuristics of GA, ACO, and Particle Swarm Optimization (PSO) to determine the optimal stacking sequence composite laminate. Based on the results of this comparison study, ACO found to outperform GA and PSO algorithms in the field of stacking sequence design (SSD). This remarkable performance of ACO in solving such NP-hard combinatorial optimization is expected where it designed to solve discrete optimization problems [2].

An investigation of five different ACO variants in the field of composite laminate stacking sequence design has been carried out in the current study. Besides, a new optimization approach has been proposed to solve this problem. The new approach uses an improved version of a well-known ACO algorithm variant called Hyper Cube Framework ACO (HCFACO). A popular NP-hard combinatorial optimization problem of composite laminate stacking sequence has been considered to demonstrate the new algorithm performance [1]. Furthermore, a performance assessment criterion has been developed using different measures, such as performance rate, reliability, normalized price, and Fitness–Distance correlation. The proposed EHCFACO algorithm performance has been compared with other ACO algorithm variants.

The rest of this paper is structured as follows: firstly, it presents a review of standard ACO followed by a brief description of other considered ACO variants; secondly, the new optimization approach is explained in detail; then, the performance assessment criterion is introduced, followed by the numerical experiments section; next, the results of the performance survey are discussed; finally, the paper concludes with a summary of the study's research contributions, limitations, and prospective directions for future research.

2. Ant Colony Optimization Algorithms (ACOs)

Dorigo [3] developed the basic Ant Colony Optimization (ACO) algorithm, or Ant System (AS), which is a metaheuristic approach inspired by the collaborative work of ants in finding the source of food. The ants cooperate to find the best possible path from their colony to the food source. During their searching tour the ants communicate by depositing a certain amount of a substance, called the pheromone, on their way to the food site. The following group of ants tends to follow the paths with higher pheromone concentration. Over time, the less selected paths gradually lose their information due to the pheromone evaporation process.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 August 2021 doi:10.20944/preprints202108.0573.v1

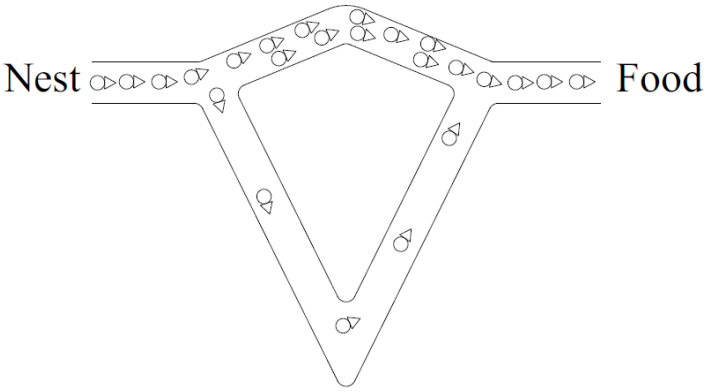


Figure 1. Cooperative search of ants by pheromone trails

The virtual ants travelling and selecting paths can be interpreted as a probabilistic selection of certain nodes, which they are part of the solution, in the path based on the pheromone value. The ACO general procedure is illustrated in **Algorithm 1**.

Algorithm 1. Ant Colony Optimization procedure

Initialization
While (termination criteria not satisfied) **Do**
Construct Solutions Table by Ants
Local Search (optional)
Global Pheromones Updating
End ACO algorithm

To understand the mathematical interpretation of ACO, there is a need to go through each step of the ACO procedure shown in **Algorithm 1**. ACO starts with initial values of the pheromone trail τ_0 , set to a small value for all ant trails as this gives all nodes j of the design variable i, an equal probability of selection. Next, each ant starts to construct its own solution by applying the rule of selection, which has the following general form:

$$p_{ij}^{(k)} = \frac{\tau_{ij}^{\alpha}.\eta_{ij}^{\beta}}{\sum_{j\epsilon N_i^k}\tau_{ij}^{\alpha}.\eta_{ij}^{\beta}}\ ,\ \forall\, j \in N_i^k \tag{1}$$

$p_{ij}^{(k)}$ represents the probability of selecting the path $i\,j$ for the k^{th} ant, τ_{ij} is the updated pheromone trail, η_{ij} denotes the value of heuristic information for each feasible solution s, $N_i^{(k)}$ indicates the neighbourhood nodes of the k^{th} ant, when it located at node i and α, β are the amplification parameters for pheromone trials and the influence of heuristic information on the algorithm behaviour respectively [12]. At the end of each tour all the pheromone trails are updated through two steps of pheromone evaporation and depositing, according to the following formula:

$$\tau_{ij}^{k(t+1)} = (1-\rho).\tau_{ij}^{k(t)} + \sum_{k=1}^{n} \Delta\tau_{ij}^{k(t)} \tag{2}$$

where ρ is the evaporation rate, $\rho \in (0,1]$, and $\Delta\tau_{ij}^{k(t)}$ is the amount of deposited pheromone by ant $k(t)$ that could be determined as:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 August 2021 doi:10.20944/preprints202108.0573.v1

$$\Delta\tau_{ij}^{k(t)} = {Q}/{L_{k(t)}} \tag{3}$$

where Q is a constant and $L_{k(t)}$ represents the distance travelled by ant $k(t)$. Equation (3) is the basic form of the pheromone trail updating which used to solve TSP optimization problem and it could be implemented in more general form:

$$\Delta\tau_{ij}^{k(t)} = \begin{cases} \xi.\frac{f_{worst}}{f_{best}}, & if\ i,j\ \in global\ best\ solution \\ 0, & otherwise \end{cases} \tag{4}$$

where f_{worst} , f_{best} are the worst and the best values of the objective function f obtained by N ants in tour t and x is the global pheromone scaling factor [13]. Eventually, the ACO loop continues until one of the termination conditions is met.

In SDD optimization problem, the thickness of each ply (equivalent to the distance between the cites in TSP) is assumed to be constant, so the heuristic information value, h, will be constant all over the ant tours t which simplifies the probability of selection, in Equation (1), into:

$$p_{ij}^{(k)} = \frac{\tau_{ij}^{\alpha}}{\sum_{j\epsilon N_i^k}\tau_{ij}^{\alpha}}, \qquad \forall\, j \in N_i^k \tag{5}$$

Local search

The procedure of the ACO algorithm includes the option of improving the intensification feature of the ACO algorithm by adding some local search algorithms or movements that could improve the search of the solution neighbourhood [2] .

2.1 Elitist Ant Colony (EACO)

Gambardella and Dorigo [5] introduced an improved version of the ACO algorithm that uses the elitism strategy. The idea behind this strategy is a reinforcement of the best solution path found once the algorithm is initialized. The rule of pheromone updating for EACO is written as follows:

$$\tau_{ij}^{k(t+1)} = (1-\rho).\tau_{ij}^{k(t)} + \sum_{k=1}^{n}\Delta\tau_{ij}^{k(t)} + e.\Delta\tau_{ij}^{k(t_{best})} \tag{5}$$

$$\Delta\tau_{ij}^{k(t_{best})} = \frac{f_{best}}{\Sigma_{k=1}^{n} f_i} \tag{**Error! No text of specified style in document.**.6}$$

Where:

The reinforcement of selection probability of the best path (t_{best}) occurs by adding the value of $e.\Delta\tau_{ij}^{k(t_{best})}$ where e is the weighting parameter and it represents the number of elitist ants [10].

2.2 The Rank-Based Ant Colony Optimization (RBACO)

Bullnheimer, Hartl [14] proposed a new extension of the ACO that enhances the performance of the original EACO by ranking the ants based on their path length. The deposited value of pheromone decreases according to its rank index, m. Moreover, only the best ants, s, will be updated which prevents the over concentration of pheromones on local optima paths chosen by other ants . Hence, the pheromone updating rule of RBACO is:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 August 2021 doi:10.20944/preprints202108.0573.v1

$$\tau_{ij}^{k(t+1)} = \rho.\tau_{ij}^{k(t)} + \sum_{\mu=1}^{\sigma-1} \Delta\tau_{ij}^{\mu} + \sigma.\Delta\tau_{ij}^{k(t_{best})} \quad (7)$$

2.3 Max-Min Ant Colony (MMACO)

Previous ACO algorithms used the strategy of reinforcing only the best-found paths. This strategy could cause the excessive increase of pheromone values on optimal local paths causing all other ants to follow this path. To overcome this drawback, Stützle and Hoos [15] proposed a modified version of ACO that limits the pheromone values to a specific interval, $[\tau_{min};\ \tau_{max}]$. In addition, the initialization of pheromone value is set to the upper limit of the pheromone interval, with a small evaporation rate to increase the algorithm search diversification. The pheromone rule is:

$$\tau_{ij}^{k(t+1)} = \rho.\tau_{ij}^{k(t)} + \Delta\tau_{ij}^{k(t_{best})} \quad (8)$$

and $\tau_{ij}^{k(t)} \in [t_{min};\ t_{max}]$

where $[\tau_{min};\ \tau_{max}]$ values are determined by the following formulas:

$$\tau_{max} = \frac{1}{(1-\rho)}.\frac{f_{worst}}{f_{best}}$$

$$\tau_{min} = \frac{\tau_{max}.\left(1-\sqrt[n]{P_{worst}}\right)}{\left(\frac{n}{2}-1\right).\sqrt[n]{P_{worst}}}$$

where p_{best} denotes the probability of the best solution, it has a value greater than 0, while n represents the number of ants.

2.4 Best-Worst Ant Colony (BWACO)

Zhang, Wang [16] presented BWACO as an extension of MMACO, where the algorithm exploitation capability benefits from both, best and worst solutions. During the search tour, the pheromone trail update uses the positive return of the best solution and the negative one generated by the worst solution. The pheromone updating rule can be written as:

$$\tau_{ij}^{k(t+1)} = \left[\rho.\tau_{ij}^{k(t)} + \Delta\tau_{ij}^{k(t_{best})}\right]_{\tau_{min}}^{\tau_{max}} - \lambda.\Delta\tau_{ij}^{k(t_{best})} \quad (9)$$

where, λ is a coefficient that has value within $[0,1]$ interval and it could be noticed that BWACO became MMACO if $\lambda = 0$.

2.5 Hyper Cube Framework ACO (HCFACO)

The different algorithms of ACO build a limited hyperspace of the pheromone values. The Hyper Cube Framework of ACO algorithms, proposed by Blum in 2001, generates a binary convex hull hyperspace from pheromone values for the feasible solutions. In other words, the values of the pheromone vector, $\tau = [\tau_1, \tau_2, \tau_3, \ldots., \tau_n]$, are limited to the interval $[0,1]$, and this is carried out by changing the pheromone update rule. The following formula expresses the rule of pheromone updating in HCFACO:

$$\tau_{ij}^{k(t+1)} = (1-\rho).\tau_{ij}^{k(t)} + \rho.\sum_{k=1}^{n} \Delta\tau_{ij}^{k(t)} \quad (10)$$

where:

$\Delta\tau_{ij}^{k(t_{best})} = \frac{f_{best}}{\sum_{k=1}^{n} f_i}$, and n is the number of ants follow the same best path.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 August 2021 doi:10.20944/preprints202108.0573.v1

HCFACO algorithms overcome the undesirable problem of different behaviour of standard ACO algorithms when the same objective function is scaled, which affects the algorithm robustness. Also, it reduces the search effort and improves the algorithm search diversification [2] . Lastly, it is worthwhile to mention that the HCF update rule is not limited to standard ACO algorithm (or Ant System AS) as it can also be used with MMACO, where the maximum and minimum limits of MMACO pheromone trail are set to be 0 and 1 respectively [17].

3. Enhanced Hyper Cube ACO Algorithms

Dorigo experimentally observed that using local search techniques can improve the overall performance of the ACO [2, 18]. Local search can be carried out by hybridizing the ACO with local search algorithms such as Tabu search or using permutation operators to explore the solution neighbourhood [9, 19]. The commonly used operators in SSD optimization problem are two-points permutation and swap. Two-points permutation means selecting two bits in the solution string and reversing the order of the bits in between [1]. The swap operator is used to switch the position of two randomly selected bits of the solution string [20].

The HCFACO algorithms presented here adopted two other permutation operators to perform the algorithm enhancement. The first operator is called a single point mutation, which is used successfully with Permutation Genetic Algorithm [1]. The second operator is inspired by using one of the Tabu Search movements named the insertion [2]. The proposed Enhanced HCFACO procedure for standard ACO (Ant System AS) and max-min ACO is listed in **Algorithm 2**.

Algorithm 2. Enhanced HCFACO procedure

Initialization

- ACO parameters:$[n_{Ants}, Iter_{max}, N_V, \alpha, \rho, \tau_0]$
- Convergence rate counter: $[Iconv_{max}]$
- initialize ACO loop counter: $[ige = 0]$

While (termination criteria not satisfied) **Do**

$$ige = ige + 1$$

- Solution table construction:

$$P_i = \tau^{\alpha}$$

$$Si(n_{Ants}, N_V) = RouletteWheelSelection(\frac{P_i}{\sum P_i})$$

- Solution evaluation:

$$f(ige, Si(1 : n_{Ants}, N_V))$$
$$S^*(ige) = max(S_i(ige))$$
$$if\ S^*(ige) < max(S^*(1 : ige - 1))$$
$$S^*(ige) = max(S^*(1 : ige - 1))$$
end

- Apply Local Search:
  - Single point mutation.
  - Insertion.

$$if\ S^*_{LS}(ige) > S^*(ige)$$
$$S^*(ige) = S^*_{LS}(ige)$$
$$end$$

Global pheromone trail updating: Eq. (10)

End (While)

Optimal solution $S_{opt} = max(S^*(1 : ige))$

End Enhanced HCFACO algorithm

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 August 2021 doi:10.20944/preprints202108.0573.v1

The Enhanced HCFACO Algorithms starts by defining the standard ACO parameters such as the maximum number of iterations $(Iter_{max})$, number of ants (n_{Ants}) , number of design variables (N_V), the initial pheromone trail (t0) and evaporation rate (r). In addition, the solution convergence rate counter is imposed $[Iconv_{max}]$ and its value determine whether the convergence rate is slow or fast. When the ACO loop starts, all solution edges have the same deposited pheromone trail τ_0 , which gives all nodes the same probability of selection to be a part of the feasible solution. The artificial ants, $k = 1{:}\, n_{Ants}$, start building the solution table, $S_i(n_{Ants}, N_V)$, by randomly choosing a node di on their way to build the solution vector $S_i(k, N)$. Next, the evaluation of the solutions table is carried out by calling the objective function, and the obtained values are stored in $f(ige, S_i(1{:}\, n_{Ants}, N_V))$ matrix. The best solution of the stacking sequence design has the maximum value of the objective function listed in $f(ige, S_i)$ matrix of the current iteration. The best solution of each iteration ige is stored in the best solution matrix $S^*(ige)$. Thereafter, the global pheromone trail update is performed as described in the Hyper Cube Framework of ACO in Equation (10).

The local search actions are enforced as soon as the best solution of the current tour is determined. Following this, a comparison of the generated solutions with the best solution obtained so far is made. The best solution matrix is then updated if any improvement is detected. Finally, the HCFACO loop continues until the termination criteria are met. The global optimal solution is determined as the best solution matrix member with the maximum value of the objective function.

4. Performance Evaluation

The time required by an algorithm to find the global optima is widely used to evaluate its performance [21]. However, a single performance measure cannot reflect the effectiveness of the algorithm in exploring the design space or determining solution quality. In the current study, three different groups of performance measures have been applied to ensure a fair evaluation of the proposed algorithm.

4.1. Computational Effort

In addition to the elapsed time, literature has shown that other measures can be used to measure computational effort. The first is the Price PS, which is defined as the number of objective function evaluations within a search run and 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. Practical optima is defined as the solution with 0.1% error in the best possible solution [1]. The last is the normalized price nPS ,which is defined as the ratio of price and practical reliability [1, 22, 23]. Finally, the Performance Rate measure Prate, is also considered to link the computation effort with the number of function evaluations [21].

$$P_{rate} = \frac{number\ of\ successful\ runs}{\begin{pmatrix} number\ of \\ function \\ evaluations \end{pmatrix} . \begin{pmatrix} total\ number \\ of\ runs \end{pmatrix}} \tag{11}$$

4.2. Solution Quality

The solution quality of an algorithm can be measured by determining the absolute error between the current solution and the best-known global solution [1, 22, 23].

$$Q = \left| \frac{S^* - S_{opt}}{S_{opt}} \right| . 100 \tag{12}$$

4.3. Fitness Landscape Analysis

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 August 2021 doi:10.20944/preprints202108.0573.v1

The design space of a combinatorial optimization problem can significantly affect the search performance of an algorithm. The notion of Fitness-Landscape appeared in literature as an answer to the question of "what the design space looks like?". The Fitness-Landscape is defined by the feasible solutions set, the objective function (fitness) and the structure of the solution neighbourhood. To find the connection between the Fitness Landscape and the problem hardness, Jones and Forrest [24] introduced a Fitness Landscape - Distance Correlation (FDC) to determine the hardness of optimization problems to be solved using Genetic Algorithm (GA). The distance mentioned here is defined as the number of movements that should be imposed on a Solution S_i to eliminate dissimilarity with the optimal solution S_{opt} . The proposed correlation by Jones is computed using the correlation factor, r :

$$r(F,D) = \frac{CFD}{\sigma_F . \sigma_D} \tag{13}$$

where CFD indicates the $Cov(F,D)$ and σ_F, σ_D are the standard deviation of F and D respectively. The values of the correlation coefficient r are limited to interval $[-1,1]$ where negative values are desirable for maximization and indicate better searching performance. Finally, using the scattering of fitness versus the distance to the global optima can reveal valuable information about FDC of the optimization problem solved by an algorithm [15, 23].

5. Numerical Experiments

To demonstrate the performance of the new approach we selected a well-know NP-hard combinatorial optimization problem in filed of composite laminated structures. The optimization objective is maximizing the critical buckling load of composite laminated plate exposed to bidirectional compression loading. The decision variables are the fiber orientation of each composite layer (lamina) which form the optimal stacking sequence of the laminate (a group of layers). To employ ACO as an optimization algorithm for SSD optimization problem, there is a need to understand specific problem characteristics such as solution representation, constraints, and objective function formulation. In meta-heuristic algorithms, the solution (stacking sequence) takes the form of a bit string that consists of a combination of plies with the available angle fiber orientations (e.g. $0°, \pm 45°$ and $90°$).The different solutions have integer coding with 1,2 and 3 numbers, which represent the three possible fiber orientations, respectively. For instance, the laminate with $[2132231]_s$ stacking sequence describes the laminate of $[\pm 45,\ 0_2, 90_2, 45, \pm 45, 90_2, 0_2\ ]s$ fiber orientations.

The simplicity of using an integer representation along with significant performance gains, made it the most widely used method in meta-heuristic optimization algorithms for composite laminated design. The buckling load factor lb for simply supported rectangular laminated plate subjected to bi-axial loading is determined as follows:

$$\lambda_b(\mathrm{p},\mathrm{q}) = \pi^2 \frac{\left[D_{11}\left(p/a\right)^4 + 2(D_{12}+2D_{66})\left(p/a\right)^2 + D_{22}\left(q/b\right)^4\right]}{\left(p/a\right)^2 N_x + \left(q/b\right)^2 N_y} \tag{14}$$

where D_{ij} denotes the bending stiffness, N_x is the axial loading in x-direction, N_y is the axial loading in y-direction, p and q are the number of half waves in x, y directions. The critical buckling load factor λ_{cb} is defined as the minimum obtained value of $\lambda_b\ (p,q)$. The critical values of p and q are linked to different factors such as laminate material, a number of plies, loading conditions and the plate aspect ratio. In uniaxial loading and simply supported plate, the critical buckling load is happening when $p = 1$ whereas in biaxial the critical buckling loads it needs to be determined as the minimum value of $\lambda_b(p,q)$ [1, 10]. Finally, the constraints in stacking sequence optimization with constant laminate thickness t could be imposed as follow:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 August 2021 doi:10.20944/preprints202108.0573.v1

- 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 $0, \pm 45$;and 90.
- Only $N/4$ ply orientations are needed to describe laminate as a result of balancing constraints.
- Contiguity constraint is handled by using the penalty parameter (β).
- And the critical buckling load factor objective function f_{obj} could be formulated as:

$$f_{obj} = (1 - \beta).\max\left(\lambda_{cb}(p, q)\right) \tag{15}$$

To compare the performance of the proposed algorithm alongside the other ACO algorithms; we implemented all the algorithms presented here using MATLAB R2019b software. The benchmarking problem from the literature of stacking sequence design optimization is accredited to Le Riche and has been used by previous studies [1, 20]. The original problem describes a simply supported plate subjected to an in-plane biaxial loading as shown in

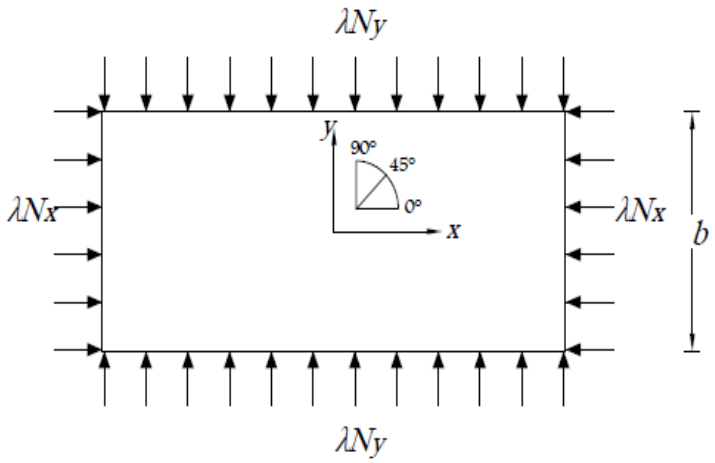


Figure 2.

Figure 2. Simply supported plate subjected to biaxial loading

The thickness of each ply t_i is assumed constant, and the ply orientations are limited to $0, \pm 45$ and 90 sets of angles. The number of plies N_L is constant. The required properties, dimensions, and loading conditions are listed in Table 1 and **Table 2**. The objective function is set to maximize the critical buckling load. The constraints are integrated into the solution (e.g., balanced laminate, symmetrical, etc.). The implemented ACO algorithms were executed on the same computer for the same number of experiments; $N_{exp} = 200$. This number is used to overcome the stochastic behaviour of meta-heuristic algo-

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 August 2021 doi:10.20944/preprints202108.0573.v1

rithms [1]. Furthermore, this number of experiments is conducted over ten different random generating seeds of $301, 2,50,75, 111, 200,167, 225$,11 and 25. Then the average of the performance measures values was used in the comparison of different ACO algorithms.

Table 1. Graphite-epoxy lamina's properties

$E_1(GPa)$	$E_2(GPa)$	$G_{12}(GPa)$	v_{12}
127.59	13.03	6.41	0.3

Table 2. Graphite-epoxy lamina's geometrical and loading data

N_L	$t(mm)$	$a(mm)$	$b(mm)$	$N_x(N/m)$	N_x/N_y
64	0.127	508	254	175	1

Lastly, all ACO algorithms were examined at two different levels of convergence rate, slow and fast. The slow rate enforces the algorithm searching loop to stop after 56 iteration without improvement, while the fast rate needs just 10 iterations to be terminated [1].

5.1. ACO Parameters Setting

To ensure a fair assessment of the ACO algorithms performance, the following standard ACO parameters were assumed for all implemented algorithms: number of Ants $n_{Ants} = 25$, the maximum number of iterations $Iter_{max} = 1000$, evaporation rate $r = 0.1$, the parameter of the pheromone trail relative importance $\alpha = 1$, initial one trail $\tau_0 = 0.004$ (except for MMACO and BWACO algorithms were $\tau_0 = 1$). Best solution probability $P_{best} = 0.05$ for MMACO and BWACO Algorithms and lastly the coefficient of worst solution pheromone trail $\lambda = 0.6$ for BWACO algorithm only.

5.2. Termination Criteria

All algorithms will stop as soon as one of the following conditions are satisfied:

- If there is no improvement in the solution after 10 or 56 iterations.
- If the number of iterations exceeds 150 and the best solution is equal to the worst solution (means all artificial ants following the same path).
- If a maximum number of iterations have been generated.

6. Results

The case study described in the previous section has been optimized using nine different algorithms: standard ACOA, EACO, RBACO, MMACO, BWACO, HCF/EHCF for both ACO and MMACO algorithms. Analysis of the algorithm's performance will be divided into two parts. First, the performance of ACO algorithms with Hyper Cube Framework will be assessed. The second part is dedicated to the comparison of EHCFACO algorithm with the rest of ACO algorithms.

6.1. Hyper Cube Framework ACO Algorithms Results Analysis

Referring to section 0, the Hyper Cube Framework (HCF) can be applied for both versions of the standard ACOA and MMACO. Hence, this part of the analysis is devoted to determining which version of both ACO algorithms, with HCF and EHCF, could exhibit better performance? The performance measures for the original ACOA,MMACO, HCFACO, HCFMMACO, EHCFACO, and EHCFMMACO are listed in

 doi:10.20944/preprints202108.0573.v1

Table 3. The performance values listed in

 doi:10.20944/preprints202108.0573.v1

Table 3 reveal that applying HCF to the ACOA has positively affected the overall performance of ACO. The average practical reliability increased by $22-56\%$ and the normalized price declined from 51.17 to 36.91 for fast convergence rate and from 181.12 to 94.24 for slow one. The performance rate doubled at slow rate while remain the same for the fast. The FDC correlation coefficient r decreased slightly for both levels of convergence.

Further improvement of HCFACO performance is acquired when the proposed local search movements are imposed. The average practical reliability became more than twice of the standard ACO and the normalized price decreases more to hold at 28.92 instead of 51.17 and 81.49 instead of 181.2 for both convergence rates. The performance rate is slightly increased and the FDC correlation coefficient r is partially improved. On the contrary, HCFMMACO performed poorly compared to the original MMACO. However, the performance of MMACO has improved when the Enhanced HCF is applied, but the computational effort became more costly. Based on these results, we conclude that EHCFACO delivers an inexpensive solution with significant performance. Eventually, the solution convergence of the algorithms mentioned above has been plotted **Figure 3**, for the same selected experiment (seeds=75).

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 August 2021 doi:10.20944/preprints202108.0573.v1

Table 3. The performance measures of Hyper Cube Framework ACO algorithms

Performance measure	Convergence rate	ACOA	HCFACO	EHCFACO	MMACO	HCFMMACO	EHCFMMACO
Elapsed time, t_s, min	Slow	0.87	.01	2.16	1.12	0.96	1.34
	Fast	3	4.15	6.86	4.62	6.68	4.39
Reliability %	Slow	35.71	77.1	89.6	16	13.17	54.36
	Fast	36.45	93.15	98.95	87.7	91.05	98.25
Normalized price, nP_s	Slow	51.17	36.91	28.92	152.23	169.26	52.2
	Fast	181.12	94.24	81.49	118.53	117.3	98.25
Performance rate, P_{rate}	Slow	0.0196	0.0272	0.0347	0.0068	0.0063	0.0206
	Fast	0.0056	0.0106	0.0123	0.0088	0.0090	0.0106
Quality %	Slow	99.69	99.93	99.96	98.28	98.57	99.81
	Fast	99.69	99.97	99.98	99.96	98.57	99.98
Fitness-Distance Correlation, r	Slow	-0.80	-0.67	-0.79	-1	-1	-0.82
	Fast	-0.84	-0.71	-0.73	-0.82	-0.77	-0.75

6.2. Other ACO Algorithms Results Analysis

All ACO algorithms have been successfully found the best-known value of the maximum critical buckling load factor, λ_{cb}. A different samples of optimal SSD obtained by different ACO algorithms were listed in

Algorithm	Optimal stacking sequence design	Critical buckling load factor, λ_{cb}
EACO	$[2333332333323333]_S$ $[\pm45°/90°_{10}/\pm45°/90°_8/\pm45°/90°_8]_S$	3973.01
RBACO	$[2333323333333332]_S$ $[\pm45°/90°_8/\pm45°/90°_{18}/45°]_S$	
BWCACO	$[3333232323222222]_S$ $[90°_8/\pm45°/90°/\pm45°/90°/\pm45°/90°\pm45_2]_S$	
EHCFACO	$[3333322322232222]_S$ $[90°_{10}/\pm45°_2/90°/\pm45°_3/90°/\pm45°_8]_S$	

. The solution convergence of EACO,RBACO and BWACO for a selected experiment (seeds=75) has been graphically illustrated in Figure Error! **No text of specified style in document.**.4.

It is observed from

Algorithm	Optimal stacking sequence design	Critical buckling load factor, λ_{cb}
EACO	$[2333332333323333]_S$ $[\pm45°/90°_{10}/\pm45°/90°_8/\pm45°/90°_8]_S$	3973.01
RBACO	$[2333323333333332]_S$ $[\pm45°/90°_8/\pm45°/90°_{18}/45°]_S$	

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 August 2021 doi:10.20944/preprints202108.0573.v1

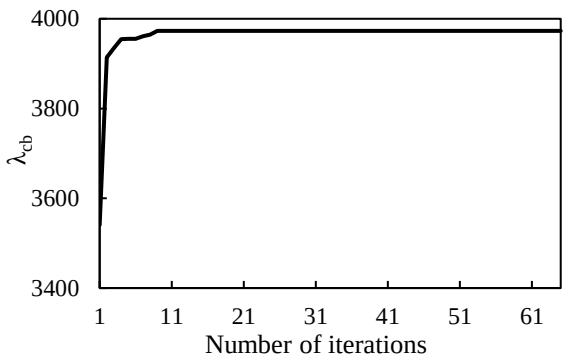
BWCACO	$[3333232323222222]_s$ $[90°_8/\pm45°/90°/\pm45°/90°/\pm45°/90°\pm45_2]_s$
EHCFACO	$[3333322322232222]_s$ $[90°_{10}/\pm45°_2/90°/\pm45°_3/90°/\pm45°_8]_S$

that the optimal stacking sequence design followed the same pattern of switching between two groups of 90_2 and ±45 fiber orientations which confirm the results of previous studies[1, 20].

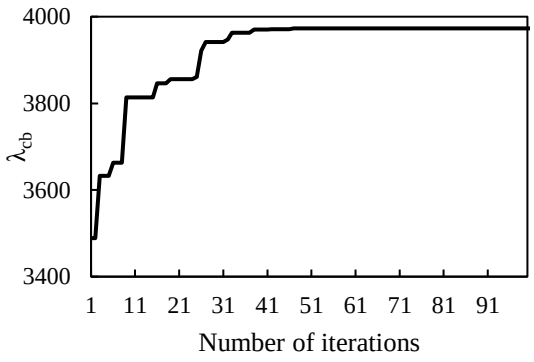
Table 4. The optimal stacking sequence for 64 ply laminates subjected to biaxial loading without contiguity constraint ($N_y = N_x = 1\ and\ a/b = 2$)

Algorithm	Optimal stacking sequence design	Critical buckling load factor, λ_{cb}
EACO	$[2333332333323333]s$ $[\pm45°/90°_{10}/\pm45°/90°_8/\pm45°/90°_8]_s$	3973.01
RBACO	$[2333323333333332]_s$ $[\pm45°/90°_8/\pm45°/90°_{18}/45°]_s$	
BWCACO	$[3333232323222222]_s$ $[90°_8/\pm45°/90°/\pm45°/90°/\pm45°/90°\pm45_2]_s$	
EHCFACO	$[3333322322232222]_s$ $[90°_{10}/\pm45°_2/90°/\pm45°_3/90°/\pm45°_8]_S$	

On the other hand, the solution convergence plot in Figure Error! **No text of specified style in document.**.4 illustrate that both RBACO and BWACO algorithms develop gradual search trends on their way to the optima whereas EACO and EHCFACO algorithms smoothly converge to the global optima. Further, the numerical experiments confirm the fluctuation of ACO algorithms in finding the global optimal solution due to their stochastic nature, as illustrated in Figure 5.



(a) ACOA



(d) MMACO

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 August 2021 doi:10.20944/preprints202108.0573.v1

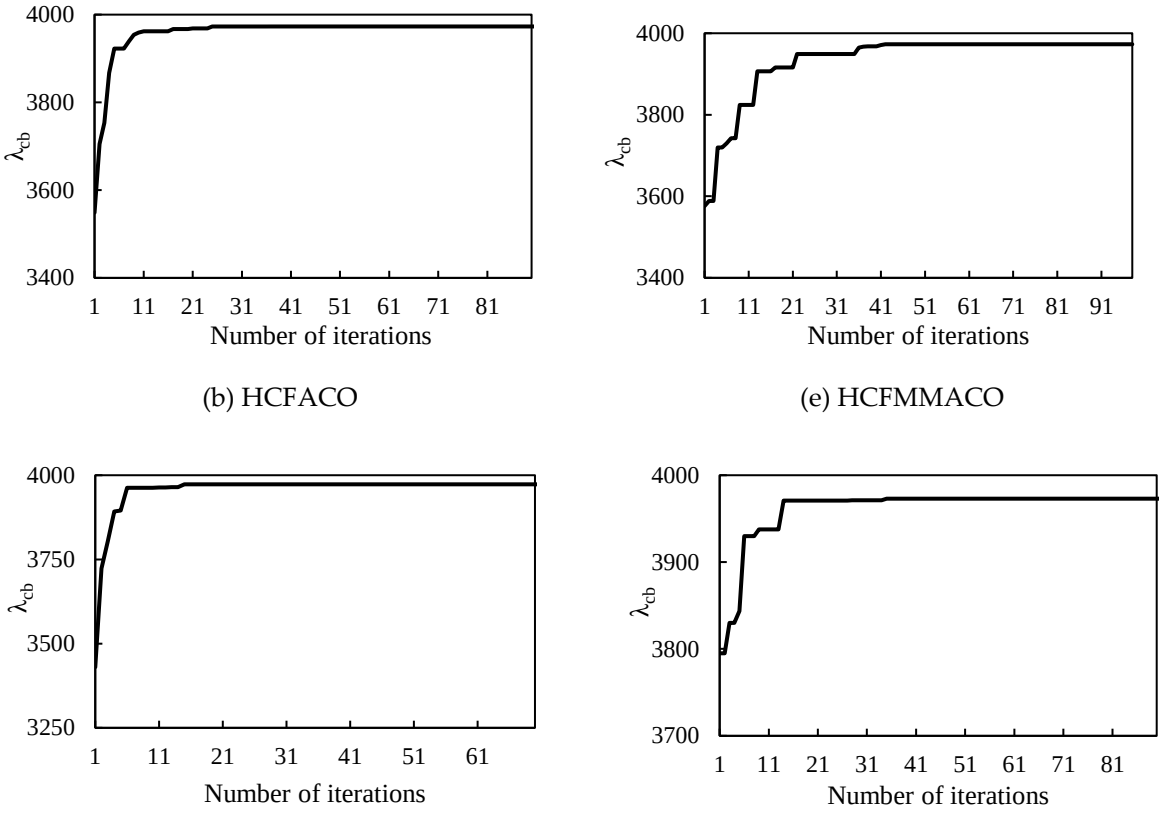
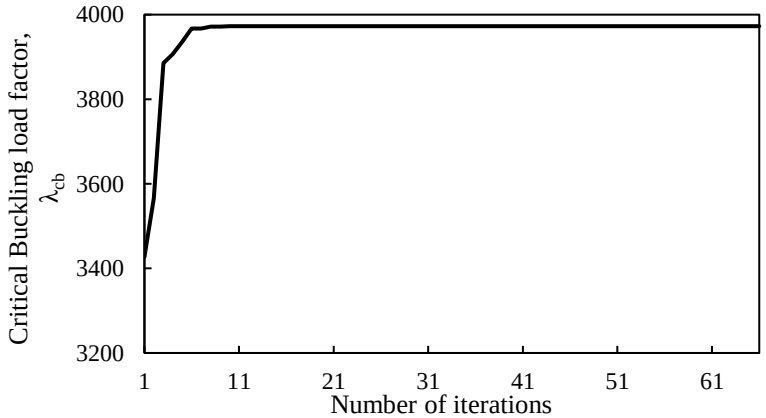


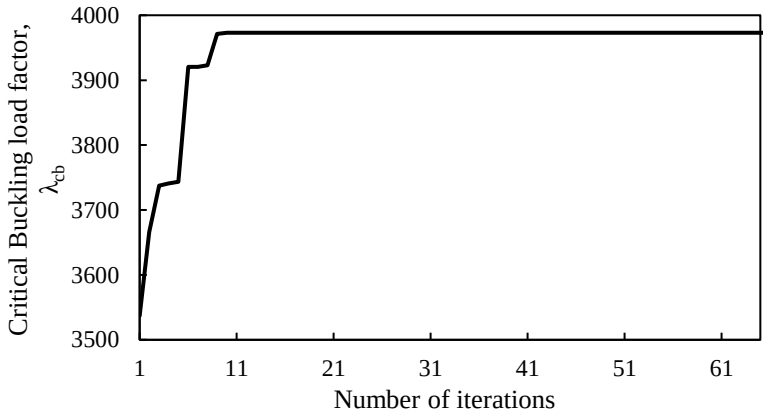
Figure 3. Solution convergence of ACO-MMACO and their Hyper Cube Framework variants

According to the introduced performance assessment criteria in Section 0, the average values of different performance measures of reliability, performance rate, solution quality, normalized price, and searching effort coefficient are determined for EACO, RBACO, BWACO at fast and slow convergence rates. These results, alongside with EHCFACO results, are plotted in Figure 6 to **Figure 12** to provide a sensible comparison of the performance evaluation of the proposed algorithm with other ACO algorithms. The average practical reliability of the algorithms is introduced in **Figure 6**. Both EACO and RBACO algorithms show low practical reliability values, with 10% and 8% respectively, while BWACO presented better value at the slow rate of convergence but it poorly performed at the fast rate. The EHCFACO algorithm exhibited a significant reliability values of 89.6-98.95%. Furthermore, it demonstrated the highest performance rate measure (0.012-0.035), see **Figure 9**. The solution quality results of the algorithms are summarized and depicted in **Figure 10** which reveals that all ACO algorithms produce an excellent solution quality for this particular case study.

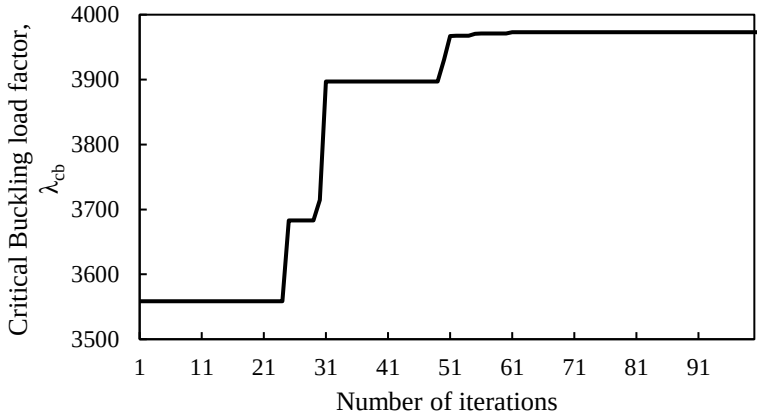


Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 August 2021 doi:10.20944/preprints202108.0573.v1

(a) EACO



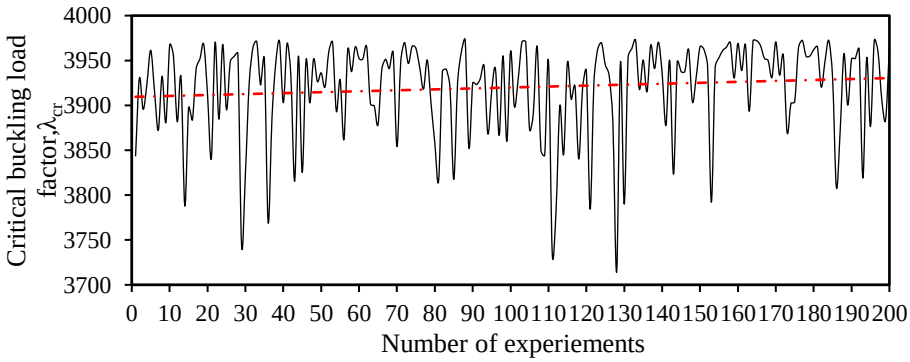
(b) RBACO



(c) BWACO

Figure **Error! No text of specified style in document.**.4 Solution convergence of EACO, RBACO and BWACO

In terms of solution cost, the normalized price results plotted on the scatter chart that illustrated in **Figure 7**. As mentioned before, the normalized price measure reflects the balance between the solution cost and the reliability. So, it is quite clear that EHCFACO outperformed other ACO algorithms. BWACO comes second at slow convergence rate, whereas RBACO and EACO deliver a costly solution.



(a) EACO

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 August 2021 doi:10.20944/preprints202108.0573.v1

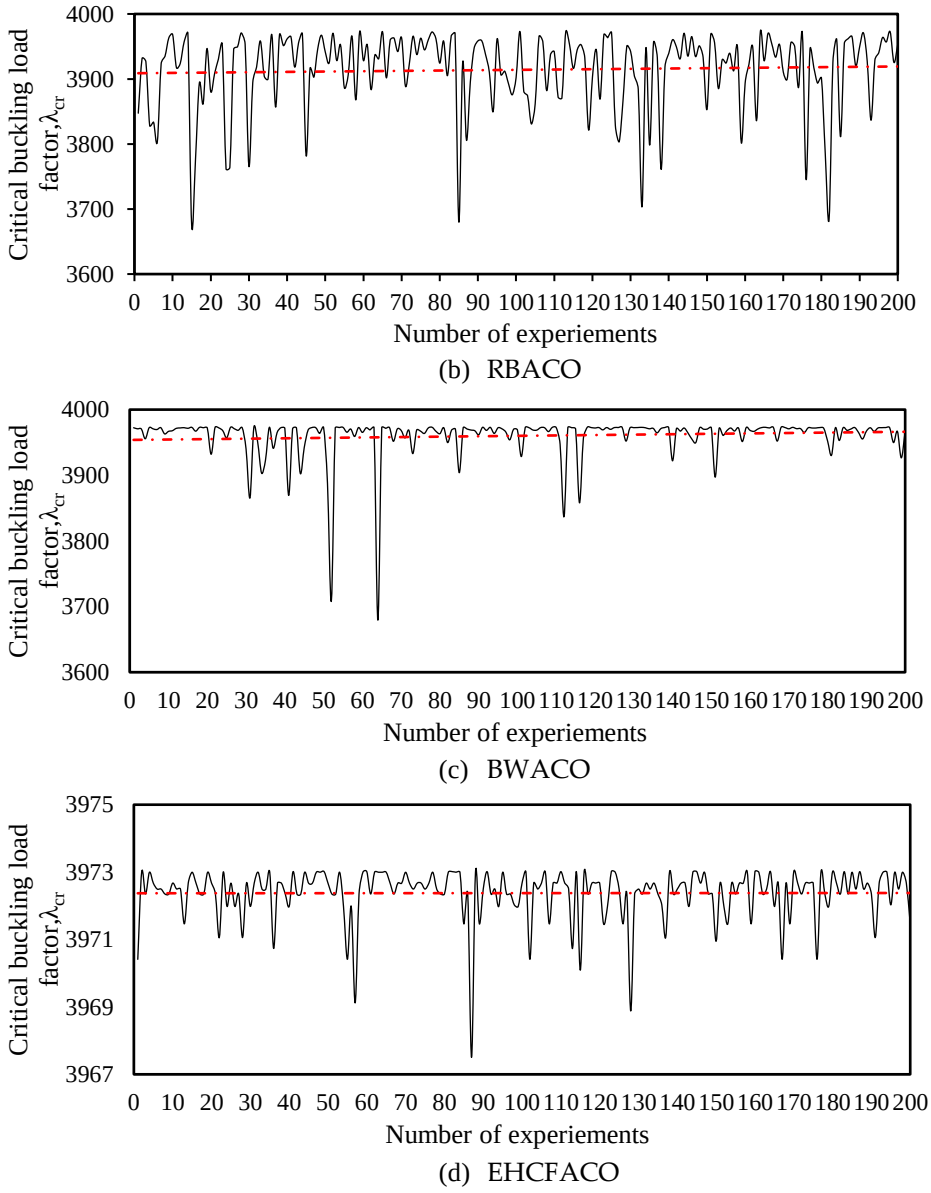


Figure 5. Critical buckling load factor vs. number of experiments

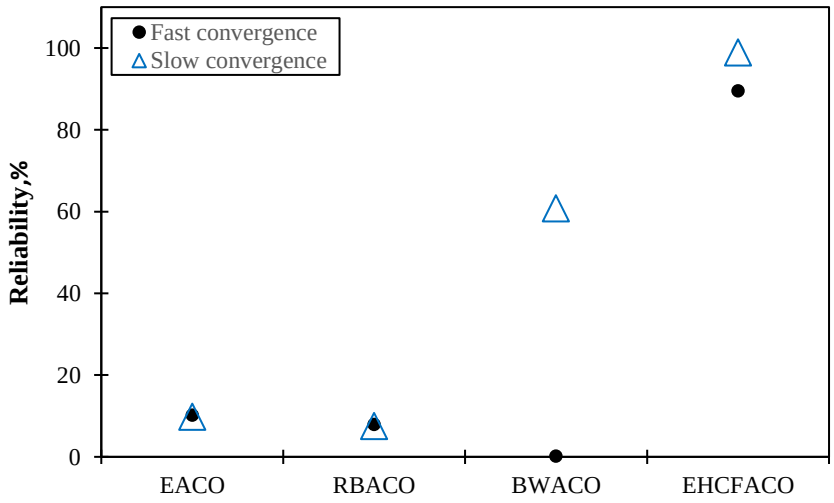


Figure 6. Reliability of EACO, RBACO, BWACO and EHCFACO algorithms

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 August 2021 doi:10.20944/preprints202108.0573.v1

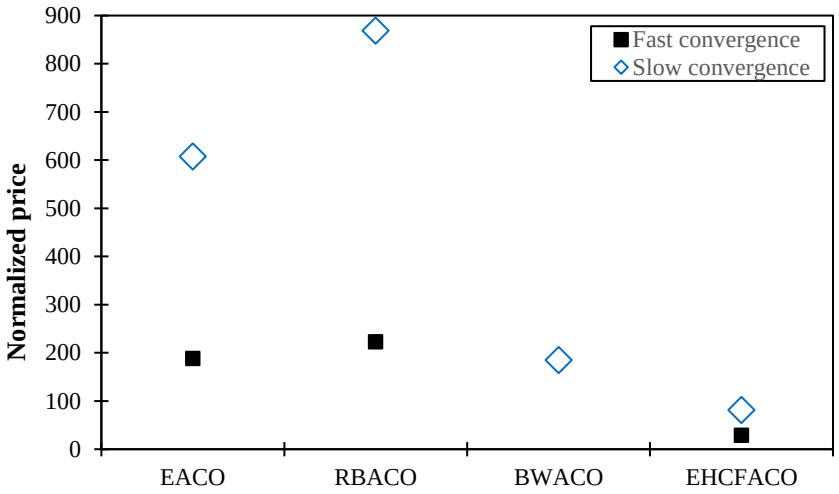


Figure 7. Normalized price of EACO, RBACO, BWACO and EHCFACO algorithms solutions.

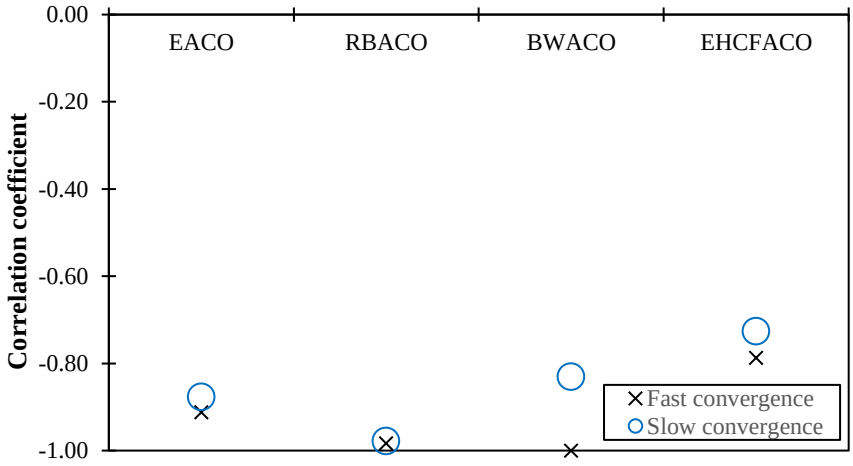


Figure 8. Correlation coefficient of EACO, RBACO, BWACO and EHCFACO algorithms.

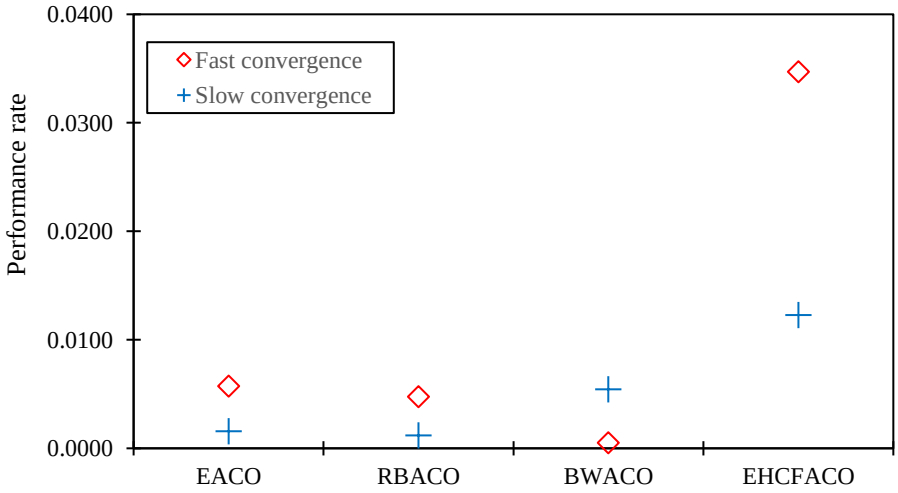


Figure 9. Performance rate of EACO, RBACO, BWACO and EHCFACO algorithms.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 August 2021 doi:10.20944/preprints202108.0573.v1

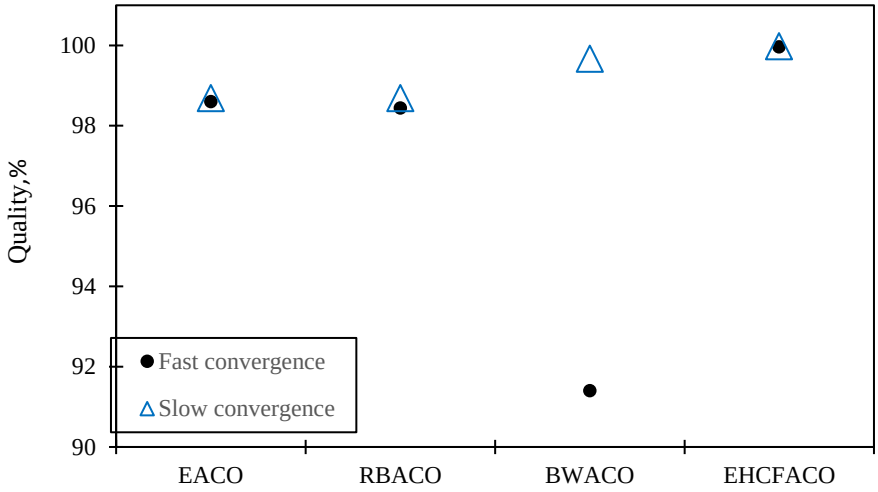


Figure 10. Solution quality of EACO, RBACO, BWACO and EHCFACO algorithms.

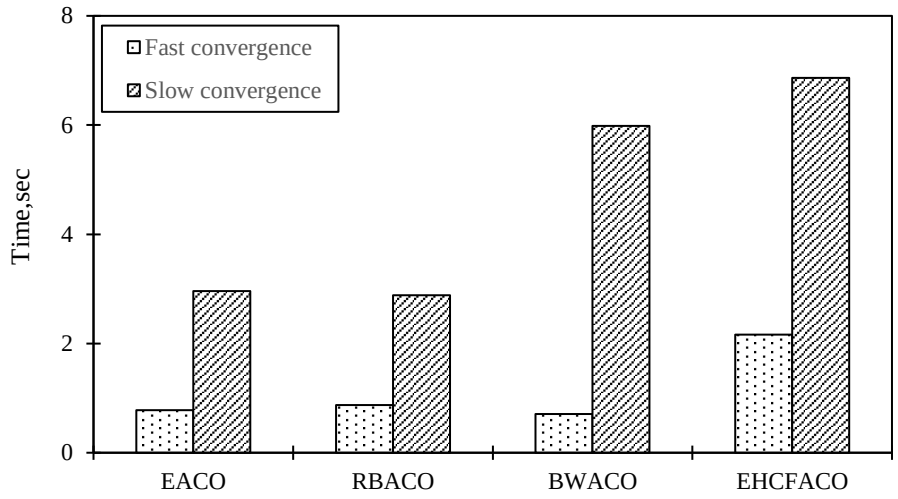


Figure 11. Elapsed time of EACO, RBACO, BWACO and EHCFACO algorithms to find the optimal solution.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 31 August 2021 doi:10.20944/preprints202108.0573.v1

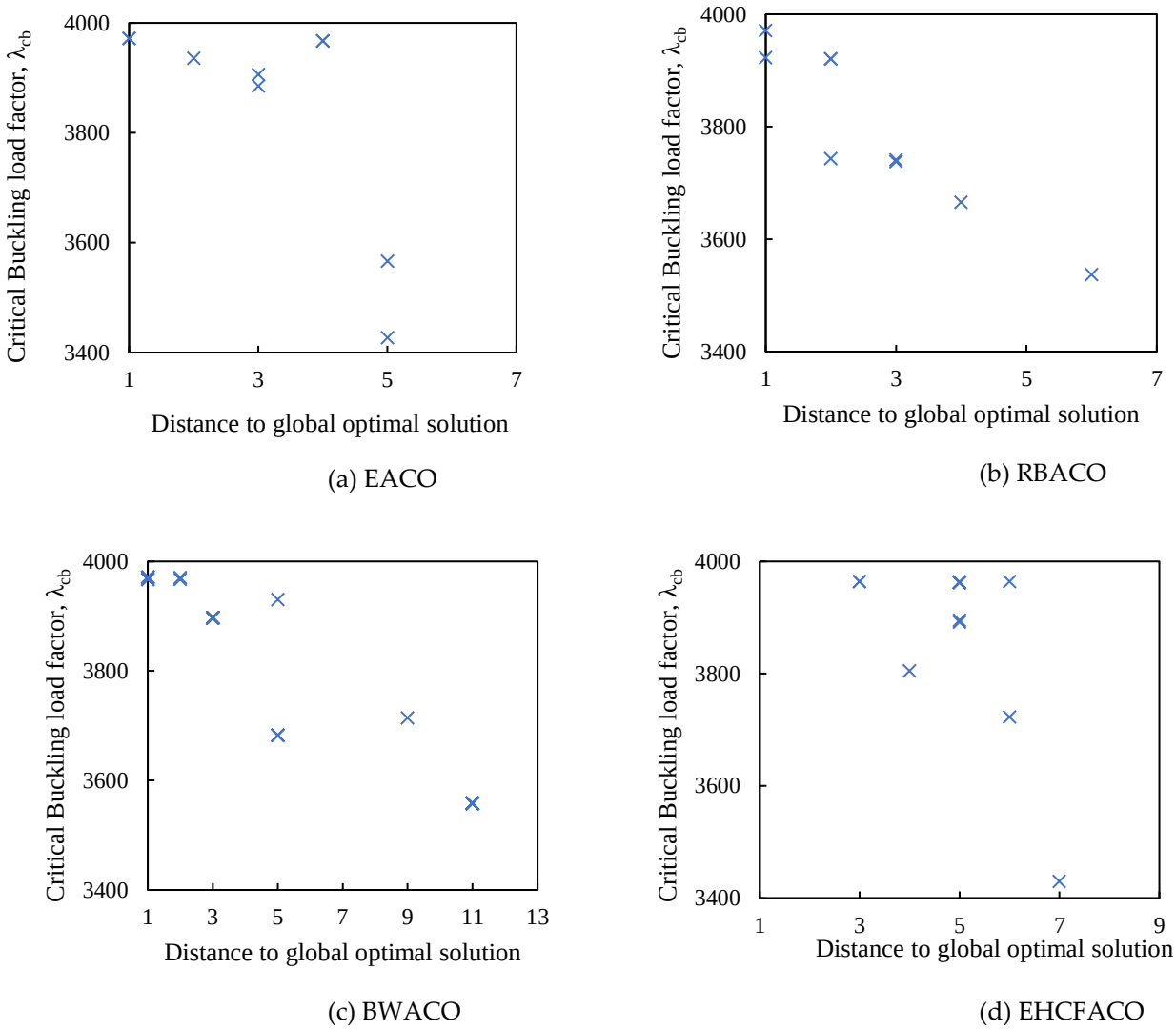


Figure 12. Fitness VS. Distance to global optimal solution of ACO Algorithms

7. Conclusion

Since many of the structural optimization problems are hard to solve within the polynomial computational time (NP), this study introduces a new optimization approach to solve the structural combinatorial optimization problems. The new approach uses an enhanced version of Hyper Cube Framework ACO (EHCFACO) that integrates two movements of insertion and bit flip to improve the local search feature of the original algorithm. A well-known benchmark case study of a composite laminated plate subjected to bi-directional buck ling loads has been selected to investigate the performance of the proposed algorithm. Furthermore, five different ACO variants were concisely presented and implemented to solve the same case study. General performance assessment measures, such as reliability, normalized price, . . . etc., have been determined for all presented ACO algorithms. It is observed that applying Hyper Cube Framework (HCF) to standard ACO has a significant influence on the overall performance of ACO. Furthermore, imposing local search movements, as an enhancement of exploitation effort, helped HCFACO to deliver a cost-effective solution. These improvements in ACO performance are in line with suggestions made by previous studies that rewarding HCF and local search movements the dominant factor in improving standard ACO algorithm performance [12, 25]. In general, the proposed EHCFACO outperforms the other ACO variants, where it offers a cost-effective solution.

Supplementary Materials: The following are available online at www.mdpi.com/xxx/s1, Data Sheets: Numerical experimental data.

Author Contributions: Conceptualization, A.A. and T.D.; methodology, A.A.; software, A.A.; validation, A.A., T.D. and V.L.; writing—original draft preparation, A.A.; writing—review and editing, T.D.; supervision, V.L.; project administration, T.D.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Le Riche, R. and R.T. Haftka, *Optimization of laminate stacking sequence for buckling load maximization by genetic algorithm.* AIAA journal, 1993. **31**(5): p. 951-956.
2. França, P.M., N.M. Sosa, and V. Pureza, *An adaptive tabu search algorithm for the capacitated clustering problem.* International Transactions in Operational Research, 1999. **6**(6): p. 665-678.
3. Dorigo, M., *Ant Colony Optimization—new optimization techniques in engineering.* by Onwubolu, GC, and BV Babu, Springer-Verlag Berlin Heidelberg, 1991: p. 101-117.
4. Weiler, C., et al. *Heuristic Approaches for the Probabilistic Traveling Salesman Problem*. in *International Conference on Computer Aided Systems Theory*. 2015. Springer.
5. Gambardella, L.M. and M. Dorigo, *An ant colony system hybridized with a new local search for the sequential ordering problem.* INFORMS Journal on Computing, 2000. **12**(3): p. 237-255.
6. Zheng, F., et al., *An adaptive convergence-trajectory controlled ant colony optimization algorithm with application to water distribution system design problems.* IEEE Transactions on Evolutionary Computation, 2017. **21**(5): p. 773-791.
7. Kaveh, A. and S. Talatahari, *An improved ant colony optimization for constrained engineering design problems.* Engineering Computations, 2010.
8. Koide, R.M. and M.A. Luersen, *Maximization of Fundamental Frequency of Laminated Composite Cylindrical Shells by Ant Colony Algorithm.* Journal of Aerospace Technology and Management, 2013. **5**(1).
9. Aymerich, F. and M. Serra, *Optimization of laminate stacking sequence for maximum buckling load using the ant colony optimization (ACO) metaheuristic.* Composites Part A: Applied Science and Manufacturing, 2008. **39**(2): p. 262-272.
10. Rama Mohan Rao, A., *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, 2009. **223**(1): p. 1-18.
11. Bloomfield, M.W., J.E. Herencia, and P.M. Weaver, *Analysis and benchmarking of meta-heuristic techniques for lay-up optimization.* Computers & Structures, 2010. **88**(5): p. 272-282.
12. Dorigo, M., M. Birattari, and T. Stutzle, *Ant colony optimization.* IEEE computational intelligence magazine, 2006. **1**(4): p. 28-39.
13. Rao, S.S., *Engineering optimization: theory and practice*. 2009: John Wiley & Sons.
14. Bullnheimer, B., R.F. Hartl, and C. Strauss, *A new rank based version of the Ant System. A computational study.* 1997.
15. Stützle, T. and H.H. Hoos, *MAX–MIN ant system.* Future generation computer systems, 2000. **16**(8): p. 889-914.
16. Zhang, Y., et al. *Best-worst ant system*. in *2011 3rd International Conference on Advanced Computer Control*. 2011. IEEE.
17. Blum, C. and M. Dorigo, *The hyper-cube framework for ant colony optimization.* IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2004. **34**(2): p. 1161-1172.
18. Dorigo, M., *Luca Maria Gambardella: ant colony system: a cooperative learning.* IEEE Trans Evol Comput, 1997. **1**: p. 53-66.

 doi:10.20944/preprints202108.0573.v1

19. Katagiri, H., et al., *A hybrid algorithm based on tabu search and ant colony optimization for k-minimum spanning tree problems.* Expert Systems with Applications, 2012. **39**(5): p. 5681-5686.
20. Jing, Z., X. Fan, and Q. Sun, *Stacking sequence optimization of composite laminates for maximum buckling load using permutation search algorithm.* Composite Structures, 2015. **121**: p. 225-236.
21. Talbi, E.-G., *Metaheuristics: from design to implementation*. Vol. 74. 2009: John Wiley & Sons.
22. Kogiso, N., et al., *Genetic algorithms with local improvement for composite laminate design.* Structural optimization, 1994. **7**(4): p. 207-218.
23. Malan, K.M. and A.P. Engelbrecht, *Fitness landscape analysis for metaheuristic performance prediction*, in *Recent advances in the theory and application of fitness landscapes*. 2014, Springer. p. 103-132.
24. Jones, T. and S. Forrest. *Fitness Distance Correlation as a Measure of Problem Difficulty for Genetic Algorithms*. in *ICGA*. 1995.
25. Dorigo, M. and T. Stützle, *Ant colony optimization: overview and recent advances*, in *Handbook of metaheuristics*. 2019, Springer. p. 311-351.