ANADOLU ÜNİVERSİTESİ

Bilim ve Teknoloji Dergisi A-Uygulamalı Bilimler ve Mühendislik
Cilt: 14 Sayı: 2 2013
Sayfa: 183-190

ARAŞTIRMA MAKALESİ /**RESEARCH ARTICLE**

İlhan KAHRAMAN[1], Abdullah Tugrul SEYHAN[1], Haluk YAPICIOGLU[2]

TEK EKSENLİ BASINÇ ALTINDA BÜKÜLME DAVRANIŞI GÖSTEREN KOMPOZİT SİLİNDİRLERİN TASARIMI VE OPTİMİZASYONU

ÖZ

Bu çalışmanın amacı kompozit silindirlerin baskı yükü altında bükülme performanslarının tahminlenmesinde genetik algoritma, kesik değerli parçacık sürü optimizasyonu ve sürekli değerli parçacık sürü optimizasyonu metodlarının karşılaştırılmasıdır. Kompozit malzemeyi oluşturan liflerin referans eksenine olan açıları tasarım değişkeni olarak ele alınırken, kompozit malzemedeki katman sayısı sabit kabul edilmiştir. Tahminlerin doğruluğu bulguların istatistiki anlamlılığı kullanılarak karşılaştırılmış ve sınıflandırılmıştır. Elde edilen sonuçlar ışığında, hem genetik algoritma, hem de kesikli parçacık sürü optimizasyonu tekniklerinin ele alınan problem için yüksek kaliteli sonuçlar ürettiği görülmüş, kesikli parçacık sürü optimizasyonu ile elde edilen sonuçların genetik algoritma kullanılarak elde edilen sonuçlardan istatistiki olarak daha iyi olduğu belirlenmiştir.

Anahtar Kelimeler*:* Lamine kompozit malzeme, İstif sıralaması optimizasyonu, Silindir, Genetik algoritma, Parçacık sürü optmizasyonu.

DESIGN AND OPTIMIZATION OF STACKING SEQUENCE FOR BUCKLING OF COMPOSITE CYLINDRICAL SHELLS UNDER UNIAXIAL COMPRESSION

ABSTRACT

This work aims at comparing the predictions obtained from various mathematical tools including genetic algorithm (GA), real valued particle swarm optimization (PSO_C) and discrete particle swarm optimization (PSO_D) for buckling of composite cylindrical shells subjected to uniaxial compressive load. Fiber orientation (stacking sequence) was considered as a design variable, while the number of plies is considered constant. The accuracy of the predictions was compared and ranked based on statistical significance of the findings. Results suggest that both GA and PSO_D is able to produce high quality solutions to the design problem, while the designs produced by PSO_D is found to be statistically better than those found by GA.

Keywords: laminated composite materials, Stacking sequence optimization, Cylindirical shells, Genetic algorithms, Particle swarm optimization.

[1] Department of Materials Science and Engineering, Anadolu University, Eskisehir, TURKEY.
E-mail: ilhankahraman@anadolu.edu.tr, atseyhan@anadolu.edu.tr
[2] Department of Industrial Engineering, Anadolu University, Eskisehir, TURKEY.
E-mail: hyapicio@anadolu.edu.tr

1. INTRODUCTION

Fiber reinforced polymer composites (FRPCs) have aroused great attention in structural applications due to their high strength to weight ratios (1-3). However, since they exhibit anisotropic mechanical properties, a more comprehensive analysis of the structure behavior is required when designing fiber reinforced polymer composites than that of metallic structures which exhibit simply isotropic mechanical properties. For this reason, in terms of engineering practice, development of new analysis methods that account for mechanical behavior of composite materials is needed (1, 2). When fibers are non-uniformly distributed in the cross-section of a composite, its mechanical and stress properties vary across the composite. Therefore, the strength and stiffness of structural elements made of composite materials depend not only on optimization of the fiber volume fraction and orientation (stacking sequence), but also on fiber distribution in the cross-section. Therefore, a laminated composite is optimized by choosing the thickness, number and orientation of the individual plies as design variables.

To achieve the best results, several optimization techniques have been developed so far. In this study, the behavior of laminated composite cylindrical shells subjected to uniaxial compressive loading was predicted through various optimization tools including genetic algorithm (GA), real valued particle swarm optimization (PSO_C) and discrete particle swarm optimization (PSO_D). The predictions obtained were compared to each other and evaluated in terms of statistical significance.

2. MODEL

Composite materials especially polymer matrix laminate composites have many parameters due to their special structure. Therefore, some assumptions are needed for calculation of their strength with precision. Fig. 1 a and b show composite laminate structures and stacking sequence configuration used in classical laminate theory, respectively while Fig 1 c depicts the model for cylindrical laminated shell.

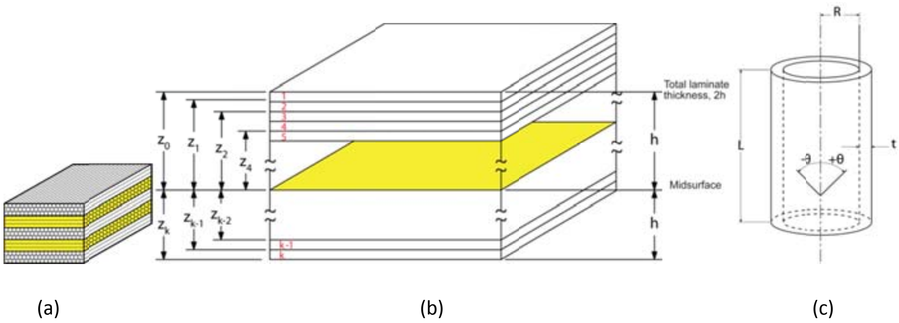

Figure 1. a) Laminated composite structure b) Laminated composite construction c) Laminated cylindrical shell.

The axial strain in a beam was related to the midplane strain and curvature of the beam under uniaxial load and bending. Similar relationships can also be developed for a plate under in-plane loads such as shear and axial forces, and bending and twisting moments. The classical lamination theory is used to develop these relationships. The following assumptions are made in the classical lamination theory to develop the relationships:

- Each lamina is elastic, orthotropic and homogeneous.
- A line straight and perpendicular to the middle surface remains straight and perpendicular to the middle surface during deformation ($\gamma_{xz} = \gamma_{yz} = 0$).
- The laminate is thin and is loaded only in its plane (plane stress) ($\sigma_z = \tau_{xz} = \tau_{yz} = 0$).
- Displacements are continuous and small throughout the laminate

- ($|u|,|v|,|w| \langle\langle |h|$), where *h* is the laminate thickness.
- No slip occurs between the lamina interfaces.

Orthotropic lamina is often considered as a material, principal directions of which do not coincide with the natural directions of the plate. This statement does not necessarily means that the material itself is not orthotropic. Rather, we are just looking at an orthotropic material in an unnatural manner, in a coordinate system that is orientated at some finite angle to the principal material coordinate system. The stress-strain relations are in the *x–y* coordinates as follows,

$$\{\sigma\} = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = [\overline{Q}]\{\varepsilon\} = \begin{bmatrix} \overline{Q}_{11} & \overline{Q}_{12} & \overline{Q}_{16} \\ \overline{Q}_{21} & \overline{Q}_{22} & \overline{Q}_{26} \\ \overline{Q}_{61} & \overline{Q}_{62} & \overline{Q}_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} \quad (1)$$

To determine the resultant moments and forces acting on the laminate, the k^{th} ply stresses, as shown in Fig. 1b, are integrated through the ply thickness. The constitutive equation for the laminate is defined as follows (3):

$$\begin{Bmatrix} \{N\} \\ \{M\} \end{Bmatrix} = \begin{bmatrix} [A] & [B] \\ [B] & [D] \end{bmatrix} \begin{Bmatrix} \{\varepsilon^m\} \\ \{\kappa\} \end{Bmatrix} = [K] \begin{Bmatrix} \{\varepsilon^m\} \\ \{\kappa\} \end{Bmatrix} \quad (2)$$

where, A, B and D matrices are as follows.

$$A_{ij} = \sum_{k=1}^{N} \left(\overline{Q}_{ij}\right)_k \left(z_k - z_{k-1}\right) \quad (3)$$

$$B_{ij} = \frac{1}{2} \sum_{k=1}^{N} \left(\overline{Q}_{ij}\right)_k \left(z_k^2 - z_{k-1}^2\right) \quad (4)$$

$$D_{ij} = \frac{1}{3} \sum_{k=1}^{N} \left(\overline{Q}_{ij}\right)_k \left(z_k^3 - z_{k-1}^3\right) \quad (5)$$

The stacking sequence lists the fiber orientations measured from a reference axis of the laminate. If the orientation is counterclockwise from the reference direction, it is considered to be negative. The standard stacking sequence lists orientations of the different layers, starting from the top of the laminate to the bottom, in a string separated by slashes. For a laminate with *N* layers, each made of the same composite material and of the same thickness, *t*, starting with the top layer with a fiber orientation θ_1, laminate is represented as (2):

$$[\theta_1 / \theta_2 / .... / \theta_N] \quad (6)$$

The thickness of each layer, *t*, in a consolidated form in the laminate is generally provided by the manufacturer's specifications. The total thickness, *h*, of the laminate is *h* = *tN*. Layers oriented at an angle from the reference axes of the laminate a called off-axis layers. When the orientation of a layer coincides with one of the reference axes of the laminate, $\theta = 0^\circ$ or $\theta = 90^\circ$, that layer is referred to as an axis layer.

When a structure (subjected usually to compression) undergoes visibly large displacements transverse to the load then it is said to buckle. Buckling may be demonstrated by pressing the opposite edges of a flat sheet of cardboard towards one another. The Euler buckling formula gives the critical load at which a long column buckles as:

$$F_{cr} = \frac{\pi^2 El}{L^2} \quad (7)$$

Where F_{cr} is the critical load for buckling, *l* is the second moment of area. It is the most basic formula for buckling. There are complex calculation approximations for laminated composites as:

$$N_{i:} = \frac{1}{\beta^2} \left\{ \widetilde{D}_{11}\beta^4 + 2(\widetilde{D}_{12} + 2\widetilde{D}_{33})\beta^2\eta^2 + \widetilde{D}_{22}\eta^4 + \frac{\left[e_{21}\beta^4 + (e_{11} + e_{22} - 2e_{33})\beta^2\eta^2 + e_{12}\eta^4 + \beta^2/R \right]^2}{a_{22}\beta^4 + (2a_{12} + a_{33})\beta^2\eta^2 + a_{11}\eta^4} \right\}. \quad (8)$$

In the context of optimization, tailoring the material properties is also associated with maximization or minimization of a performance criterion. The performance criterion may be the weight of the laminate, and the desirable response quantities, such as stiffness, become the constraints. Alternatively, a response quantity can be maximized or minimized subject to a constraint on weight. In either case, the variable typically used for design optimization defines the fiber orientation and the number of plies that make up the composite laminate. In this study, we are interested in maximizing the stiffness of the laminate composite given a constant number of layers.

Most commercially available composite materials are manufactured as a unidirectional tape with a fixed thickness. Hence, when laminate thickness is optimized, the optimal number of layers in the laminate needs to be determined which is an integer optimization problem. Despite the integer nature of the problem, most of early work in the design optimization of composite laminates was based on the use of continuous-valued ply thicknesses as design variables. This was partly due to the unavailability of easy-to-use commercial integer programming software and the high computational cost of solving integer programs.

3. OPTIMIZATION PROCEDURES

A laminated composite can be designed by choosing the thickness, number and fiber orientation of each individual ply. To achieve the best results, metaheuristic techniques has been employed (1), (2) and (3). Major advantages of using metaheuristics include the following: (i) it is not necessary to have a gradient information and it can be applied to problems where the gradient is hard to obtain or does not exist, (ii) they do not get stuck in local optima if handled properly, (iii) they can be applied to non-smooth or discontinuous functions, (iv) they furnish a set of good solutions instead of a single one. On the other hand, the use of metaheuristics exhibit some drawbacks, which include the following: (i) they require the tuning of many parameters by trial and error to maximize efficiency; (ii) a priori estimation of their performance is an open mathematical problem; and (iii) a large number of function evaluations is required to yield good results, which can make the use of metaheuristics nonviable depending on the computational cost of each function evaluation.

In this study, three different optimization methods are employed. These are genetic algorithm (GA), real valued particle swarm optimization (PSO_C) and discrete particle swarm optimization (PSO_D). To the best of our knowledge, PSO has not been used in the literature before as an optimization tool in the design of composite structures. The main motivation behind the use of PSO_C is to obtain a basis for comparison. The results obtained from the use of (PSO_C) provide an upper bound for the maximum value of bending stiffness. PSO_D, on the other hand, is examined to see the performance of the PSO on a combinatorial optimization problem. The implementations of the three algorithms used are detailed in the subsequent sections.

3.1. Genetic Algorithm

Genetic Algorithm (GA) is one of the most widely used metaheuristics within the context of combinatorial optimization problems. Theoretical framework of GA goes back to 1960s, and it is formally introduced by Holland in 1975 (2). Since then, it has been used to solve many combinatorial optimization problems successfully including but not limited to, traveling salesman problem, sequencing and scheduling, graph coloring, knapsack problems and bin packing problems. Genetic algorithms (GAs) are stochastic global search and optimization methods that originally inspired by Darwin's theory of natural selection. GAs operate on a population of potential solutions, applying the principle of survival of the fittest to produce successively better approximations to the best solution. At each generation of a GA, a new set of approximations is created by the process of selecting individuals, according to their level of fitness in the problem domain and reproducing them using operators borrowed from natural genetics.

A GA generally involves genetic operators (such as crossover and mutation) and selection operators intended to improve fitness of an initial random population. Selection usually involves a fitness function characterizing the quality of an individual in terms of the objective function and the other elements of the actual populations. Thus, a GA usually starts with the generation of a random initial population and iterates by generating a sequence of populations from the initial one. At each step the genetic operators are applied to generate new individuals. The fitness of each individual is computed and the whole population is ranked according to fitness. A subpopulation is then selected to form a new population. Many selection methods may be found in the literature. In this study, rank selection is applied. Then, all the procedure is repeated until a stopping condition is satisfied (3).

In GA, the variables of the problem at hand are represented as a vector of variables, which is also called the chromosome. Traditionally, a chromosome is a string of 0s and 1s, however, there are many other implementations as well. For instance, in this problem we used values varying from 1 to 12 as genes which map to following orientation angles for each layer of lamina.

Since the angles –90˚ and 90˚ refers to the same orientation angle, we do not need to represent them separately. Whenever a new individual is created, first the solution is converted into angles using the values from Table 1 and then the fitness of the individual is calculated by equation (8).

Table 1. Solution Representation and Decoding

Gene	1	2	3	4	5	6	7	8	9	10	11	12
Angle	-75	-60	-45	-30	-15	0	15	30	45	60	75	90

In GA implementation, an initial population of size n = 30 is created randomly. For parent selection, binary tournament selection is used. First, two individuals from the parent population are randomly selected. If one of the individuals has a better fitness than the other, then that individual is selected as one of the parents. The same procedure is repeated for the selection of the next parent. Two offspring are created from each parent pair using single point crossover with a crossover rate of 0.8. Next, the offspring are mutated with a probability of 0.01. The mutation operator is a simple swap, where two randomly selected members of the solution are interchanged. With these settings, an offspring population of n individuals is created at every generation. To select the next generation of individuals, the offspring population replaces the parent population. However, the best two individuals from the parent population are retained. The search terminates after 1,000 generations. The same number of generations is used in PSO variants as well to allow the GA to run for the same number of function evaluations as that of PSO.

3.2. Particle Swarm Optimization

Particle Swarm Optimization (PSO) was first proposed by Kennedy and Eberhart (6), Kennedy (7) and Kennedy and Eberhart (8). They were inspired by the collective behavior of bird flocks. In such populations, individuals determine their travel path considering both their previous experience (cognition) and the collective behavior of the entire flock (social interaction). PSO is a population based optimization method in which individuals (e.g. particles) use the principles of flocking in search of a best solution of an optimization problem. In PSO, each particle is composed of three vectors (current location, **c**, current velocity, **v**, and the best solution found by the particle, **p**) and two fitness values (*c-fitness* and *p-fitness*, that represent the particle's current and the best value of the objective function, respectively). At each iteration of the algorithm the current position of the particle is updated according to equations below:

$$\begin{aligned} v_{id} &= K\left(v_{id} + \varphi_1 \mathrm{U}(0,1)\left(p_{id} - c_{id}\right) + \varphi_2 \mathrm{U}(0,1)\left(p_{gd} - c_{id}\right)\right) \\ c_{id} &= c_{id} + v_{id} \end{aligned} \qquad (9)$$

where $\mathbf{c}_i$ is the current position of particle i, $\mathbf{v}_i$ is the current velocity of particle i, $\mathbf{p}_i$ is the best solution identified by particle i, $\mathbf{p}_g$ is the best solution in the neighborhood of particle i, φ_1 and φ_2 are the learning rates governing the cognition and social interaction within the swarm, respectively, d is the d^{th} dimension of the corresponding vector, d = 1, 2, … D, U(0, 1) is a Uniform random number in the interval [0, 1],and K is the constriction coefficient controlling the velocity update to prevent explosion and provide stability as explained in (9). Once **c** is updated, *c-fitness* is calculated, if it is better than *p-fitness*, **p** and *p-fitness* are also updated. In this fashion, particles never die; they just explore the search space based on the information provided to them from their own previous experience and the experience of the whole swarm. This is the most important distinction of PSO from GA. As stated earlier, we used two variants of PSO in this study, namely the continuous PSO (PSO_C) and the discrete PSO (PSO_D). Implementation details of the algorithms are provided below:

Continuous PSO – PSO$_C$: This approach is employed mainly to provide a basis of comparison on the performance of the other two optimization methods (GA and PSO$_C$). It is thought that relaxing the constraint on the fiber orientation angle of the layers, it would be possible to increase the value of the objective function further, which is actually the case, as will be shown later. In PSO$_C$, a swarm of 30 particles are initialized randomly and $\varphi_1 = \varphi_2 = 2.05$. The orientation angles are allowed to take continuous values from –90˚ and 90˚. Similar to GA, PSO$_C$, is allowed to run for 1,000 iterations. The pseudocode of the algorithm is as follows:

```
Procedure PSOC{
  t = 0;
  Initialize Swarm(t); //Swarm at tth
  cycle
  Evaluate the particles of the
  Swarm(t);
  While (Not Done) {
    for each particle i ∈ Swarm(t) {
      Select the best member of the
      Swarm(t), g;
      Update (i, g);
      Evaluate particle i;
      }end for
    t ++;
    } end while
} end procedure
```

Discrete PSO – PSO$_D$: The discrete PSO is identical to PSO$_C$ except that when calculating the fitness of the current position of the particle, we first discretize the particle's current position according to a scale which is calculated by dividing the interval [-90˚, 90˚] to twelve equal sub intervals. Then the fiber orientation angle for a specific layer is determined according to these sub intervals.

```
Procedure PSOD{
  t = 0;
  Initialize Swarm(t); //Swarm at tth
  cycle
  Evaluate the particles of the
  Swarm(t);
  While (Not Done) {
    for each particle i ∈ Swarm(t) {
      Select the best member of the
      Swarm(t), g;
      Update (i, g);
      cD ← ci; //Calculate the
      discretized angle values for
      particle i;
      Evaluate particle i using cD;
      }end for
    t ++;
  } end while
} end procedure
```

As can be understood from the pseudocode, PSO$_D$ actually works on real valued parameters, and only for objective function evaluation purposes we discretize the values of the decision variables.

4. RESULTS & DISCUSSION

In this study, we consider a laminate composite cylindrical shell which is coposed of ten layers. In order to provide a better comparison of the optimization methods chosen, we worked with two different types of materials: Carbon Epoxy and Glass Epoxy. Thus, we have two test problems in total. The optimization problem is defined as the maximization of buckling load subject to maximum number of ten layers and orientation angles of multiples of 15˚ within the closed interval of [-90˚, 90˚]. The former constraint is handled through the solution representation and the latter is handled through lower and upper bounds on decision variables, which are defined as the orientation angles of the layers. Note that the PSO$_C$ does not enforce the latter constraint, that is, with this procedure orientation angles of the layers can take any value between -90˚, and 90˚. We performed 30 independent runs with each procedure and performed a statistical analysis to compare the results. In order to make a fair comparison, each procedure is run for the same number of function evaluations which is 30,000. Fig. 2 and Fig. 3 shows the box plot of the results obtained through three algorithms for the test problems. From Fig. 2 and Fig. 3, it can be said that the PSO$_C$ yields the best performance on average in maximizing the buckling load, followed by PSO$_D$ and GA. However, the constraint on the orientation angles is relaxed for PSO$_C$. The results of the F-test performed (Table 2 and Table 4) reveals that the difference between the average performances of the three algorithms is significant at 95% level of confidence.

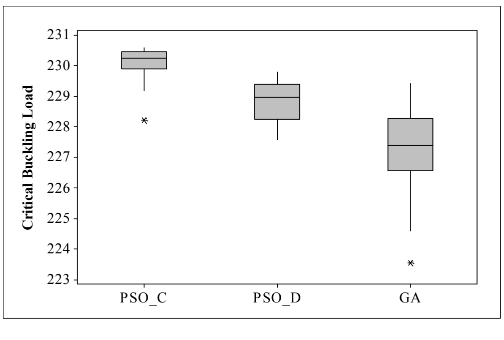

Figure 2. Boxplots of PSO$_C$, PSO$_D$ and GA: Carbon Epoxy

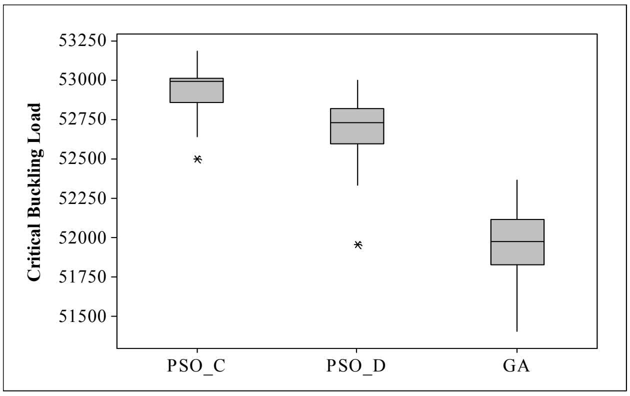

Figure 3. Boxplots of PSO_C, PSO_D and GA: Glass Epoxy

Table 2. ANOVA Table: Carbon Epoxy

Source of Variation	*SS*	*df*	*MS*	*F*	*P*	*F crit*
Between Groups	119.41	2	59.7	59.4	0.000	3.101
Within Groups	87.46	87	1.0			
Total	206.87	89				

Table 3. Statistical Analysis: Comparison Between Means: Carbon Epoxy

	Ave.	Max.	st.dev.	*t*	*P*	*t crit*
PSO_C	230.1	230.6	0.526			
PSO_D	228.8	229.8	0.670	8.024	0.000	2.00
GA	227.3	229.4	1.513	5.186	0.000	2.02

Table 4. ANOVA Table: Glass Epoxy

Source of Variation	*SS*	*df*	*MS*	*F*	*P*	*F crit*
Between Groups	15158011	2	7579005	209.98	0	3.101
Within Groups	3140152	87	36094			
Total	18298163	89				

Table 5. Statistical Analysis: Comparison Between Means Glass Epoxy

	Ave.	Max.	st.dev.	*t*	*P*	*t crit*
PSO_C	52931.89	53183.39	144.6			
PSO_D	52689.32	52994.61	209.9	13.4	0	2
GA	51965.76	52364.48	208.2	5.2	0	2

We also performed two t-tests two see whether the average performance of PSO_C vs. PSO_D and PSO_D vs. GA is significant at 95% level of confidence. Table 3 and Table 5 summarize the results of these two t-tests. As stated earlier, PSO_C provides an upper bound on the maximum value of buckling load. On average, the results obtained via PSO_D are significantly lower than the upper bound. However, the difference is less than 1%. The difference between the average performance of PSO_D with that of GA, albeit smaller, is significant at 95% level of confidence for the first test problem. The difference between the average performance of PSO_D with that of GA, is more pronounced for the second test problem.

5. CONCLUSION

In this study, the design and optimization of stacking sequence of laminated composite cylindrical shells. We applied both GA and a discrete version of PSO to maximize the bending stiffness of the laminated composite using the orientation angles as decision variables. To provide a basis for comparison, a continuous version of the PSO is also considered in which the constraint on the orientation angles relaxed. Results obtained from two test problems suggest that both PSO_D and GA are able to identify good solutions, and PSO_D being the superior optimizer for the problem at hand. One likely explanation for this situation is that the PSO_D does not use the discrete values in the search procedure, however the GA does. Thus, the PSO_D performs the search for the best solution in a continuous space while the GA works in a combinatorial environment. The predictions obtained in this study will be correlated with the experimental measurements.

REFERENCES

B. Geier, M. Piening, R. Zimmermann on The Influence of Laminate Stacking on Buckling of Composite Cylindrical Shells Subjected Uniaxial Compression. *Composite Structures*. 55, 467-474, 2002

A. T. Seyhan, G. Tayfur, M. Karakurt, M. Tanoglu. Artificial Neural Network (ANN) Prediction of Compressive Strength of VARTM Processed Polymer Composites. *Computational Materials Science*. 34 99-105, 2005

Z. Kolakowski , On Some Aspects of The Modified TSAI-WU Criterion in Thin-Walled Composite Structures. Thin walled Structures. 41 357-374, 2003

J. H. Holland, *Adaptation in Natural and Artificial Systems.* University of Michigan Press, Ann Arbor, 1975.

R.H. Lopez, M.A. Luersen, and J.E. Souza, “Optimization of Hybrid Laminated Composites using a Genetic Algorithm, “Journal of The Brazilian Society of Mechanical Sciences and Engineering, Vol. 21, No: 3, 269-278, 2009

J. Kennedy, and R.C. Eberhart, “Particle swarm optimization,” *in: Proceedings of the 1995 IEEE International Conference on Neural Networks,* 1942–1948, 1995.

J. Kennedy, “The particle swarm: Social adaptation of knowledge,” *in: Proceedings of the 1997 International Conference on Evolutionary Computation, Indianapolis,* IN, 1997, 303–308, 1997.

J. Kennedy, and R.C. Eberhart, *Swarm Intelligence,* Morgan Kaufmann Publishers, San Francisco, 2001.

M. Clerc, and J. Kennedy, “The Particle Swarm—explosion, Stability, and Convergence in A Multidimensional Complex Space,” IEEE Transactions on Evolutionary Computation vol. 6. 58–73, 2002.