Pavel Y. Tabakov* and Sibusiso Moyo

# A comparative analysis of evolutionary algorithms in the design of laminated composite structures

**Abstract:** The increased use of composite materials and structures in many engineering applications led to the need for a more accurate analysis and design optimization. While methods of stress-strain analysis developed faster, optimization techniques have been lagging behind. As a result, many designed structures do not fulfill their full potential. The present study demonstrates the major achievements in recent years in an application of evolutionary algorithms to the design optimization of fiber-reinforced laminated composite structures. Such structures are of much interest due to high structural design sensitivity to fiber orientations as well as complex multidimensional discrete optimization problems. Using an anisotropic multilayered cylindrical pressure vessel and an exact elasticity solution as an example, we show how the optimum, or near–optimum, solution can be found in a more efficient way.

**Keywords:** evolutionary algorithms; laminated composites; optimization.

DOI 10.1515/secm-2014-0385
Received October 30, 2014; accepted March 20, 2015

## 1 Introduction

It has been obvious through history that the evolution of technology has been controlled by the materials available. It is increasingly so today, and composite materials are among the most demanded. The increased use of such materials and structures in many engineering applications led to the need for a more accurate analysis and design optimization. While most of the major analytical theories were already developed in the last century, there were no reliable design methods for complex laminated composites with the use of an appropriates failure criterion until the late 1980s [1]. In fact, a five-layer anisotropic structure was probably a limit of more or less accurate results. The tensile, compressive, and shear stresses may even result from simple loading conditions, and therefore, the failure mode of composite structures is rather complicated, and this is especially true for a ply optimization. With an increasing number of layers the terrain of the functional space becomes very complex. Calculus-based methods are hampered by features of the terrain such as “ridges”, “canyons”, “flat spots”, and multiple extrema.

Thus, such problems fundamentally required new approaches and techniques. With the advent of evolutionary algorithms (EA), it became possible to open up new multidimensional and complex problems for an accurate design optimization. Nature is striking in its complexity, and despite its apparent chaotic appearance, it is well ordered and follows clear rules. Most of these rules can be explained by the theory of evolution through heredity, mutation, and selection. It is, therefore, not surprising that the scientists involved in computer research, in search of inspiration, turned to the theory of evolution. The idea of using a natural selection in optimizing problems arising in modern science belongs to John Holland [2], of the University of Michigan in the mid-1970s. A bit later, the genetic algorithms (GA) were well explained and popularized by Goldberg [3]. An excellent collection of engineering design problems can be found in the book by Mitsuo and Runwei [4]. Since then, a significant number of publications have appeared on the subject.

An idea to use a number of particles that constitute a swarm moving around in the search space looking for the best solution came to a social-psychologist James Kennedy and an electrical engineer Russell Eberhart in 1995 [5]. The new technique was called particle swarm optimization (PSO). Although, the genetic algorithm greatly extended the scope of solved problems, the high dimensions of a functional (search) space remained a formidable obstacle to an effective optimization. The arrival of the PSO algorithm has greatly expanded the dimensionality of optimizing problems. It is a robust stochastic optimization technique based on the movement and intelligence of swarms, which applies the concept of social interaction to problem solving. The particles in the swarm cooperate.

***Corresponding author: Pavel Y. Tabakov,** Department of Mechanical Engineering, Durban University of Technology, Durban, South Africa, e-mail: pashat@dut.ac.za
**Sibusiso Moyo:** Institute of Systems Science, Durban University of Technology, Durban, South Africa

They exchange information about what they have discovered in the places they have visited.

The Big Bang-Big Crunch (BB-BC) algorithm, proposed by Erol and Eksin [6] in 2006, relies on one of the evolution theories of the universe, namely, the BB-BC theory. In the BB phase, the population of feature vectors randomly fills the space, while in the BC phase, these points are drawn into a dense cluster with the center of gravity being the optimum solution of the optimization problem. Another version of this approach, called "BC" optimization method, was given by Kripka and Kripka [7], where the universal gravitation law was incorporated into the algorithm. The BB-BC method has quickly demonstrated its superiority over other heuristic population-based search techniques when employed to perform structural optimization tasks, for example, for the optimal design of space trusses [8, 9]. The BB-BC algorithm is a heuristic population-based evolutionary optimization method. Among the merits of this method are computational simplicity, ability to handle multidimensional problems, and very fast convergence. However, it seems that the implementation of it can be problematic when a noisy multimodal functional space is encountered, where there are a few local minima or maxima of a similar magnitude. Fortunately, the optimization problems considered in this paper tend to have one perceptible extreme point.

The literature on the subject is quite voluminous, and thus, the list of references is not intended to be a comprehensive one, and the specific publications are referred to because of their relevance to the present paper. Among the first comprehensive contributions to the field of design optimization of laminated composite materials is a book by Gürdal et al. [10]. The genetic algorithm was probably the first evolutionary algorithm to win general acceptance and broad application [11–13]. Later on, new evolutionary algorithms were developed and came to the attention of engineers and researchers, particularly the PSO [14, 15]. However, there is limited literature available on the application of the BB-BC algorithm to the design optimization of composite laminated structures. Some recent publications on the subject that deserve attention are [16, 17].

Using the design optimization problem for the search of an optimum fiber orientation in complex laminated structures as an example, the performance of the aforementioned optimizing methods is demonstrated and subsequently discussed.

## 2 Performance vs. ply orientation

Even today, most composite designs exploit a standard symmetric set of angles such as 0°, ±30°, ±45°, ±90°, etc. As pointed in [1], this approach to design is mostly driven by the inherited conservatism of structural engineers. Often, the numerical complexity of an accurate design optimization as well as the manufacturing limitations is another reason to overlook this important issue and, instead, increase the total cost of the final product due to an excessive use of materials.

However, even a slight change of fiber orientation in laminated composites can result in a considerable change of the stress-strain state in the structure. This is especially true if a composite structure has more or less complex geometry. This phenomena is well illustrated in Figures 1 (single layer) and 2 (two layers) by the example of two

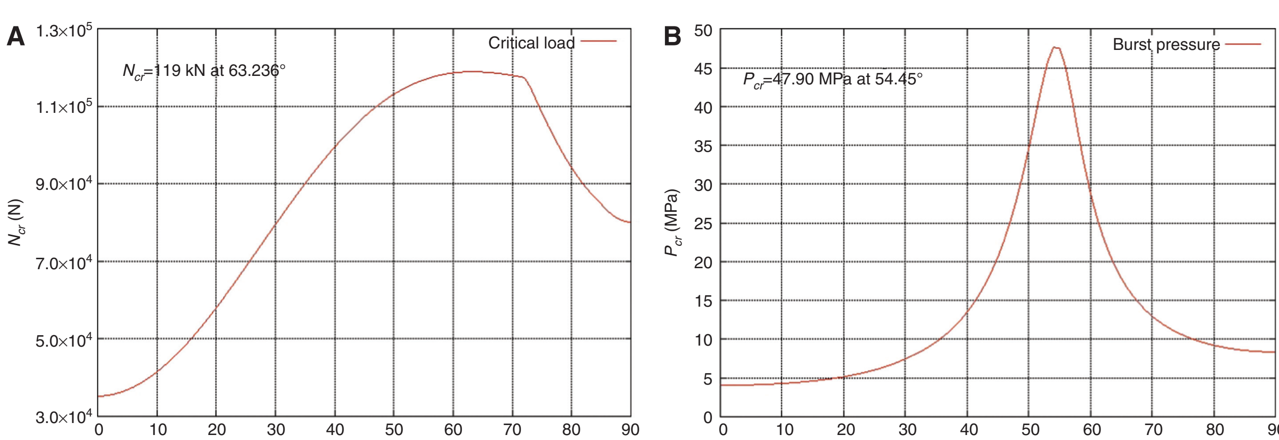


**Figure 1:** Variation of the buckling load in a rectangular plate and the critical pressure in a cylindrical pressure vessel depending on the fiber orientation. Case of a single ply.

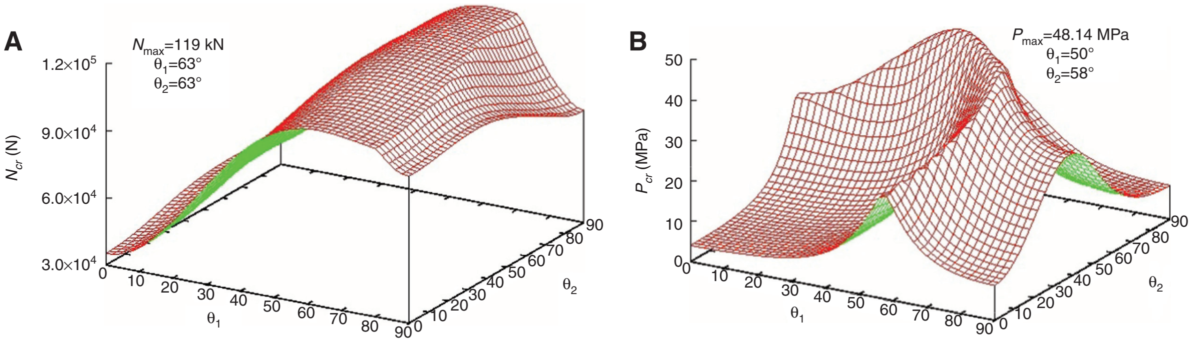


**Figure 2:** Variation of the buckling load in a rectangular plate and the critical pressure in a cylindrical pressure vessel depending on the fiber orientation. Case of two layers.

common problems. The first problem shows a variation of the buckling load in a thin-walled orthotropic plate depending on the fiber orientation. The simply supported plate lies in the Cartesian *x-y* plane with dimensions, length $a$=1.5 m, width $b$=1 m, and thickness $h$=0.01 m. The plate is subjected to a compressive force $N_x$ in the $x$ direction, and $N_y$ in the $y$ direction with the load ratio $\lambda=(N_x/N_y)=1$. A detailed description of this problem can be found in [18]. The material properties used for this and the subsequent problems are those for a typical T300/5208 graphite-epoxy material and subjected to the Tsai-Wu failure criterion [19].

The second example, the variation of the critical pressure in a cylindrical pressure vessel (exact solution), is even more illustrative. The structure under consideration is a cylindrical shell of finite length made from anisotropic material that is constructed from filament-wound layers with a fiber orientation of (±0). There are no restrictions on the number of layers, their properties, or their sequences. The axis of anisotropy coincides with the axis of symmetry of the cylinder, and the stresses act on the planes normal to the generator. The distribution of the stresses is identical in all cross sections and depends only on the radius $r$ from the axis. Therefore, the stresses can be expressed in terms of the stress functions Φ and Ψ as given in [20] where

$$\sigma_r=\frac{1}{r}\frac{\partial\Phi}{\partial r}+\frac{1}{r^2}\frac{\partial^2\Phi}{\partial\theta^2}+\bar{U},\qquad \sigma_\theta=\frac{\partial^2\Phi}{\partial r^2}+\bar{U}$$
$$\tau_{r\theta}=\frac{\partial^2}{\partial r\partial\theta}\left(\frac{\Phi}{r}\right),\ \tau_{rz}=\frac{1}{r}\frac{\partial\Psi}{\partial\theta},\ \tau_{\theta z}=-\frac{\partial\Psi}{\partial r} \tag{1}$$

and $\bar{U}$ is the potential of the body forces. The normal stress in the longitudinal direction is given as

$$\sigma_z^{(m)}=C-\frac{1}{\alpha_{33}^{(m)}}(\alpha_{13}^{(m)}\sigma_r^{(m)}+\alpha_{23}^{(m)}\sigma_\theta^{(m)}+\alpha_{34}^{(m)}\tau_{\theta z}^{(m)}) \tag{2}$$

where $C$ is the constant, which takes into account the closed ends of the pressure vessel, and the coefficients, $\alpha_{ij}^{(m)}$, are components of the compliance matrix with subscripts $i$, $j$ taking on the values as shown in (2).

A detailed mathematical apparatus based on the above approach and used for an exact elasticity analysis of the pressure vessel was developed and can be found in [21].

As a more challenging and interesting example, the pressure vessel design optimization is used next. In the case of the single-layered pressure vessel, the optimum angle is approximately 54.4°, and even a small deviation from these number results in a significant drop in performance of the structure. A similar situation is observed when two layers are used. It can be seen that the optimum angles (50° and 58°) are not the same because of the nonlinear distribution of the stresses through the thickness of the vessel. It is also interesting to note that an average value of the ply angles here is 54°.

Obviously, it is not too difficult to determine the optimum angles in the above problems. But the problem becomes increasingly challenging with an increase in the number of layers (NL). We can only guess how complex the "functional terrain" will be in these problems.

The fitness (objective) function for calculating the burst pressure is highly complex and nonlinear. The actual computer code for this function consists of about seven hundreds of lines. Taking into account a high sensitivity of the solution, the optimization procedure is performed with an accuracy of just 1°. This results in $91^{(NL)}$ different combinations of the ply orientations and can become an uncomprehendingly large number. The classical calculus-based methods cannot handle such optimization problems, and the evolutionary algorithms might be an only answer to this.

# 3 Evolutionary algorithms in search for an optimum solution

Evolutionary algorithms are population based and make use of numerical optimization. They do not require any gradient information and do not make any assumption about the function landscape. Besides, there is no direct link between algorithm complexity and problem complexity. The common basic concept of such algorithms is the survival of the fittest. There are two fundamental forces that govern an evolutionary system and propel its population. The first one is recombination and mutation, which usually employs stochastic processes and is responsible for the diversity of the population. The second one is a selection procedure for the formation of a new generation. It also often involves stochastic techniques.

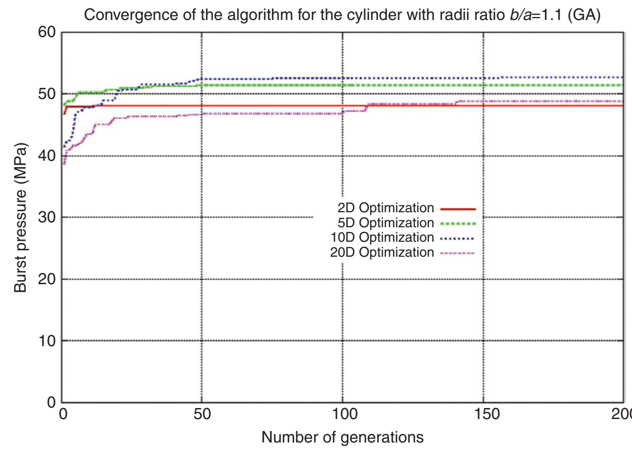


**Figure 3:** Convergence of GA algorithm for the pressure vessel with a radii ratio $b/a$=1.1.

## 3.1 Genetic algorithms

Genetic algorithms are the popular type of evolutionary algorithms. It is difficult to name a branch of science where they are not used. It is not surprising that with an advent of GA, the situation with the design optimization of composite structures has radically changed. The genetic algorithm consists of three operations: *function evaluation*, *selection*, and *reproduction*. Besides this, two main classes of genetic operations are *mutation* and *crossover*. The foundation of the GA is its *population*, and the efficiency of the algorithm directly depends on how successfully it is organized. Each *individual* contains the phenotype, the genotype, the fitness value, and some auxiliary data.

The performance of the algorithm is excellent, fast, and efficient for 1, 2, 3, 4, and 5 design parameters. However, its performance changes with the increase in the number of design parameters to 10 and 20. The length of the chromosome becomes too long to efficiently exchange the genetic material. The fitness function quickly reaches about 90% of its maximum value and slows down after that, practically without any "improvement" of the fitness function over many generations. With a careful tuning of the algorithm, the final result is achieved after about 4 h only. At the same time, there is no guarantee that the optimum result might be achieved at all. The results of the first 200 generations are shown in Figure 3. The value of the fitness function (burst pressure) is expected to increase with the increase in the number of layers. However, it can be seen that even after 200 generations (20,000 function iterations) and 10 and 20 design parameters, this is not the case. Obviously, if a high accuracy is not required, it is an excellent algorithm for multidimensional optimization as well.

Taking into account the fact that in the case of 20 design parameters there are $91^{20} \approx 1.52 \times 10^{39}$ different combinations (and remembering the complexity of the objective function), this is not bad at all. It is hardly possible to comprehend such a huge number. Its order is $10^{13}$ times larger than the diameter of the observable universe given in meters. A sample of this optimization is given in Table 1 for a medium thick pressure vessel with a ratio of an external to internal radii $b/a$=1.1 and a various number of layers each having an identical fiber orientation.

In the above optimization standard, arrangements are used; that is, the binary coding of chromosomes (seven bits per one parameter), population size is 100, selection type is tournament and/or roulette wheel, crossover probability is 80%, and the mutation rate is 2%. A detailed description of the optimization as well as extra data can be found in [21].

**Table 1:** Optimum ply angles and the maximum burst pressure obtained using the genetic algorithms.

| $b/a$ | No of layers (NL) | Optimum angle combination (inside to outside) | Burst pressure (MPa) |
|---|---|---|---|
| 1.1 | 1 | 54.5 | 47.90 |
| | 2 | 50/58 | 48.14 |
| | 3 | 48/47/64 | 48.93 |
| | 4 | 46/49/49/67 | 49.95 |
| | 5 | 46/46/46/50/75 | 51.44 |
| | 10 | 48/47/47/47/48/48/48/54/63/85 | 53.23 |
| | 20 | 46/51/46/50/49/46/45/49/47/45/ 48/49/49/51/53/57/61/65/72/87 | 53.76 |

## 3.2 Particle swarm optimization

The emergence of the PSO methodology in 1995 broke the GA dominance of many years. This methodology was designed for the optimization of nonlinear functions by using a number of particles that constitute a swarm moving around in the search space looking for the best solution. It is a robust stochastic optimization technique based on the movement and intelligence of swarms, which applies the concept of social interaction to problem solving. The swarm is moving around in the search space looking for the best solution. Each particle is treated as a point in an *N*-dimensional space, which adjusts its "velocity" according to its own flying experience as well as the flying experience of other particles. The particles in the swarm cooperate and exchange information about what they have discovered in the places they visited. The cooperation is very simple. In basic PSO, it is like this: a particle has a neighborhood associated with it and knows the fitnesses of those in its neighborhood and uses the position of the one with the best fitness. This position is simply used to adjust the particle's velocity. Having worked out a new velocity, its position is simply its old position plus the new velocity.

The main advantage of PSO over the GA is its much better ability of handling multidimensional spaces and, of course, simplicity. One particle represents one solution described by its position and velocity. The entire algorithm can be written down as follows

$$X_i=(x_1^{(i)}, x_2^{(i)}, x_3^{(i)}, \ldots, x_N^{(i)})\in\Re^N$$
$$V_i=(v_1^{(i)}, v_2^{(i)}, v_3^{(i)}, \ldots, v_N^{(i)})\in\Re^N, \tag{3}$$

where each particle maintains an individual best position (*p*Best), and a swarm maintains its global best position (*g*Best). A swarm particle *i* will update its own speed and position according to the following equations:

$$V_{i+1}=w\cdot V_i+C_g\cdot r_1(p\text{Best-}x_i)+C_s\cdot r_2(g\text{Best-}x_i)$$
$$X_{i+1}=X_i+V_{i+1} \tag{4}$$

where $w$ is a constant ("weight"); $C_g$ and $C_s$ are *cognitive* (importance of personal best) and *social* (importance of neighborhood best) *learning rates*; $r_1$ and $r_2$ are randomly generated numbers within the range {0, 1}. $V_i$ has to be limited to $V_{max}$, which must be chosen carefully; otherwise, it can be too low, too slow, too high, or too unstable.

However, the performance of PSO is very sensitive to the control parameters, which is the main disadvantage of the algorithm. It might take some time to tune these parameters in order to avoid the partial optimization ("wandering" locally) or the other way around, "flying over" the optimum point or region.

Despite the above, it is well justified to spend a bit of time adjusting the control parameters ($V_{max}$, $C_g$, and $C_s$). The result proves to be outstanding. In our case, instead of 4 h, the same problem can be solved within a couple of minutes. Moreover, the results obtained are even better than those obtained with GA. Table 2 shows the results for 10 and 20 layers.

The parameters used here are $V_{max}$=2, $C_g$=2.5, $C_s$=1.5, and the population size is 50. This worked very well for all the cases except the most complicated 20D case, where $V_{max}$=1.5. The graph shown in Figure 4 demonstrates the convergence of the algorithm. Clearly it overperforms the GA. The initial values of the fitness functions are relatively high at the early stage, and later the convergence slows down, improving little by little. It can be seen that, even in the case of the most complex optimization, the algorithm is stabilized (approaching its optimum values) before it achieves 200 generations (200×50=10,000 function evaluations).

## 3.3 Big Bang-Big Crunch algorithm

The BB-BC optimization method was proposed by Erol and Eksin [6] in 2006 as a new evolutionary algorithm.

**Table 2:** Optimum ply angles and the maximum burst pressure obtained using the PSO.

| *b*/*a* | No of layers (NL) | Optimum angle combination (inside to outside) | Burst pressure (MPa) |
|---|---|---|---|
| 1.1 | 10 | 48/46/46/45/45/46/50/55/64/90 | 53.28 |
| | 20 | 48/50/47/48/46/48/48/46/46/45/ 47/48/49/52/54/57/62/66/72/90 | 53.92 |

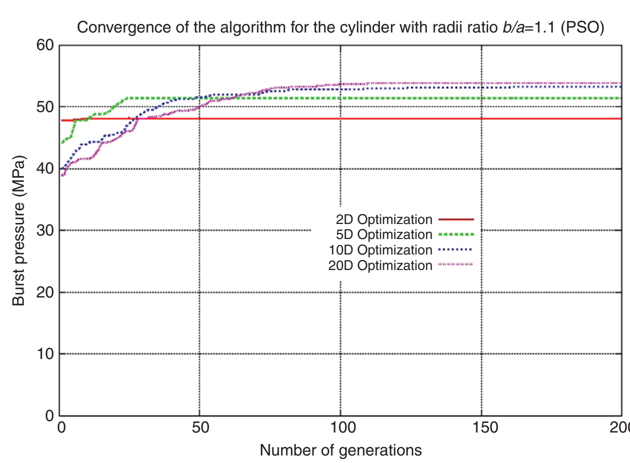


**Figure 4:** Convergence of PSO algorithm for the pressure vessel with a radii ratio *b*/*a*=1.1.

It relies on one of the evolution theories of the universe, namely, the BB-BC theory. In the BB phase, the population of feature vectors randomly fills the space, while in the BC phase, these points are drawn into a dense cluster with the center of gravity being the optimum solution of the optimization problem. The algorithm is a heuristic population-based evolutionary optimization method. Among the merits of this method are computational simplicity, the ability to handle multidimensional problems, and very fast convergence. The optimization problem can be stated as an extreme-value problem where the main objective is to find a set of parameters ($x_1, x_2, ..., x_n$), which maximizes or minimizes a quantity dependent upon them. In the present paper, by finding the maximum possible burst pressure in the pressure vessel, the objective function is maximized. As this method does not enjoy wide usage compared to the genetic algorithms and PSO, its optimization procedure is briefly outlined next.

The initial population of feature vectors is randomly generated and spread over the entire search space, allowing also some individuals (within the range of 10%) to be generated outside the search space. Then, all the points which fall outside the prescribed limits are placed at the boundaries. This will guarantee that the optimum solution point will not fall outside the domain filled in by the candidate points. The number of individuals in the population must be big enough in order not to miss the optimum point. However, the population size can be significantly reduced as the search domain shrinks.

The fitness values are computed for every individual, and in the case of maximization, the center of mass is calculated as follows

$$x_c^{(k)} = \frac{\sum_{i=1}^{N_{\text{pop}}} f_i x_i^{(k)}}{\sum_{i=1}^{N_{\text{pop}}} f_i}, \qquad k=1,\ 2,\ \ldots,\ n \tag{5}$$

where $n$ is the number of parameters, and $N_{\text{pop}}$ is the population size. The boundaries of a new contracted space are then recalculated as

$$\sigma_k = \frac{|x_{\max}^{(k)} - x_{\min}^{(k)}|}{N_{\text{gen}}+1}, \qquad k=1,\ 2,\ \ldots,\ n \tag{6}$$

where $N_{\text{gen}}$ is the generation (iteration) number. The new limits of the parameters are calculated respectively as:

$$\begin{aligned} x_{\min}^{(k)} &= \beta x_c^{(k)} + (1-\beta)\, x_{\text{best}}^{(k)} - \sigma_k \\ x_{\max}^{(k)} &= \beta x_c^{(k)} + (1-\beta)\, x_{\text{best}}^{(k)} + \sigma_k \end{aligned} \tag{7}$$

In order to control the influence of the global best solution on the boundaries of the new search space, an empirical parameter $\beta$ ($0 \le \beta \le 1$) is introduced. The new search space is now randomly filled with the points, and thus, a new population is created.

Hence, the algorithm is repeated until the stop criteria are met by setting *a priori* the total number of generations. Contrary to the other evolutionary algorithms, this number can be easily estimated from the contraction rate of the solution domain. As the search space is contracted with each new iteration, the algorithm does not walk round in circles and arrives at the optimum point very fast.

As is evident from the foregoing, the algorithm is very simple, does not require much tuning, and can easily handle much higher numbers of dimensionality than the genetic algorithms. Furthermore, the same optimization problem is solved using the BB-BC algorithm. The next four figures (Figures 5 and 6) clearly demonstrate how rapid the algorithms converge to the optimum. As an example, we consider here a three-dimensional problem (the maximum of what can be illustrated graphically). Each point in the figures represents one solution and has its own "mass" (fitness value). The population size used is 100. After about 50 generations (5000 functions evaluations) and a few seconds, the optimal result is achieved. However, an amazing feature of the algorithm is that the contraction rate of the space is the same for any number of dimensions. The multidimensional problems have more dense working space and would require a bit of higher numbers of generations before achieving the optimal or near-optimal solution. This fact is attributed to the nature of multidimensional spaces associated with high sensitivity of the numerical analysis. Nevertheless, even in the most complex case of a 20D problem, the near-optimum solution is already achieved after only 50 generations. This phenomenon is illustrated in Figure 7, which shows the convergence of the algorithm for the problems having different dimensions. Thus, instead of 4 h (using GA) and within 1 min, we managed to crunch a 20D problem ($1.52\times10^{39}$ variants of the solution) and even obtained the better final result (see Table 3).

The results obtained using the BB-BC optimization algorithm are comparable with those obtained by the PSO. They are very simple to implement and allow us to achieve a desired result incredibly fast. However, the BB-BC does not require laborious tuning. Though, the performance of the BB-BC is outstanding in this case, its applicability might be somewhat limited for other types of problems.

## 4 Nested optimization

Some design optimization problems can be even more complex by nature and require the so-called "nested

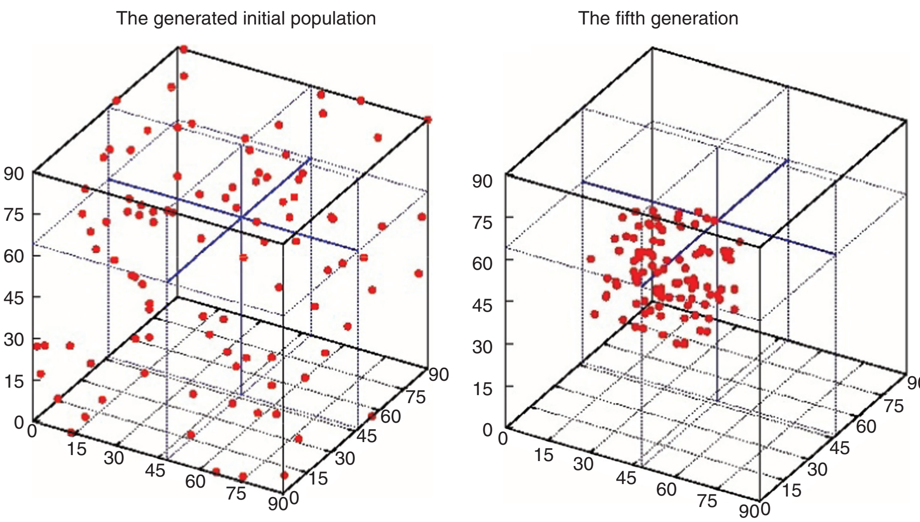


**Figure 5:** The initial population of 100 individuals and the fifth generation (BB-BC).

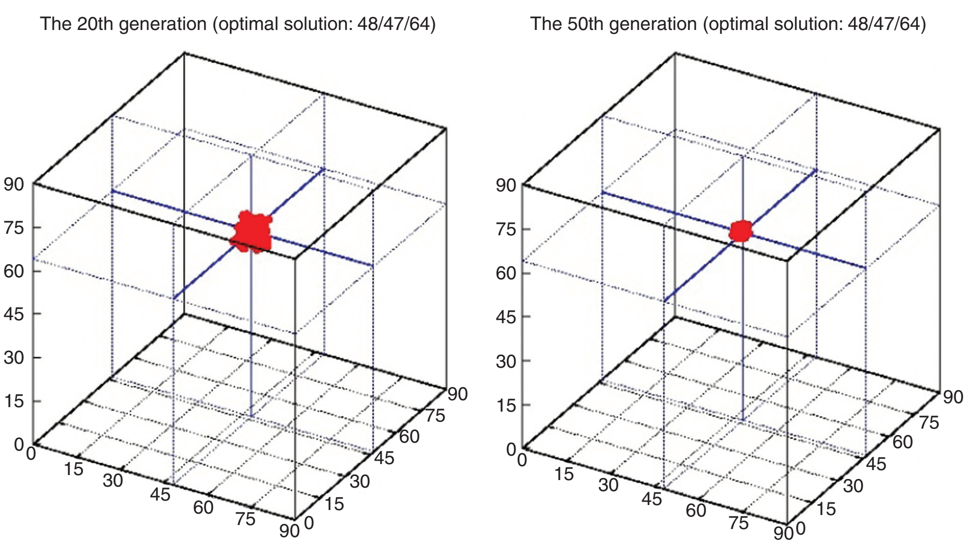


**Figure 6:** The 20th and 50th generations (BB-BC).

optimization" when one algorithm is embedded inside another one. For example, the optimal design of fiber-reinforced composite structures with manufacturing tolerances accounted for by their multidimensional nature and requires multiple steps. Although computationally expensive, nested optimization seems to be an only solution here [18]. Generally, there is a probability that every design parameter can experience deviation from its intended value in either direction thus, there are $2^N$ different scenarios for $N$ variables. The optimum solution is extremum (minimum or maximum) point on the intersection of functional (hyper-)surfaces (see Figure 8). The thick bold line in Figure 8 represents the solution line for two, three, and four tolerance surfaces. In all cases, the solution line is the line common to all the surfaces presented. As is evident, the domain of the points common to all the functional spaces, representing all possible tolerance conditions, must be determined first, and only afterward can the global optimum within this domain be established. Thus, understanding the problem of optimization with manufacturing tolerances included supports the conclusion that nesting optimization is inevitable. The detailed analysis of such problems can be found in [18].

Using the Tsai-Wu failure criterion [19], we attempt to calculate the maximum burst pressure with respect to the fiber orientations in the layers and taking into account the

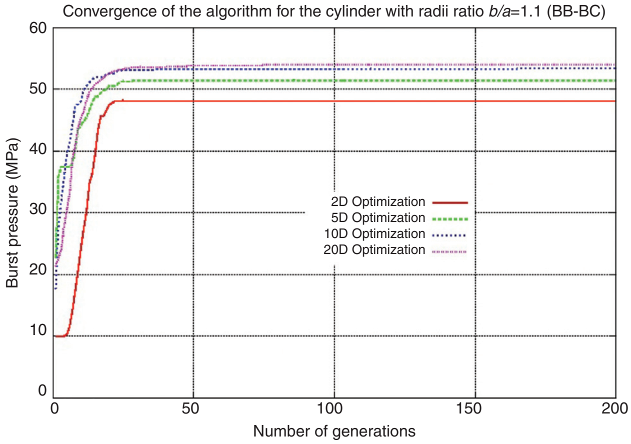


**Figure 7:** Convergence of BB-BC algorithm for the pressure vessel with a radii ratio $b/a$=1.1.

**Table 3:** Optimum ply angles and the maximum burst pressure obtained using the BB-BC algorithm.

| $b/a$ | No of layers (NL) | Optimum angle combination (inside to outside) | Burst pressure (MPa) |
|---|---|---|---|
| 1.1 | 10 | 47/46/46/46/46/46/50/55/64/88 | 53.34 |
| | 20 | 47/47/46/46/47/47/47/47/47/46/ 46/48/50/52/54/57/61/66/73/89 | 54.04 |

manufacturing tolerances. Generally, the problem can be stated as follows:

$$\max P_{cr} \stackrel{\text{def}}{=} \max_{\theta} \min_{R} P_{cr}(\theta, R). \tag{8}$$

Usually, the maximization and minimization problems are of different complexity and nature and, therefore, require different approaches. Considering the given problem, it is convenient to employ the BB-BC algorithm for the global search of the maximum burst pressure and a fast and robust microgenetic algorithm for the nested algorithm of minimization. Such a combination and cooperation of different techniques allows us to perform highly complex optimization procedures.

Here, it is required to find the worst scenario from all the possible combinations ($2^{N+1}$), then, the best result by changing the fiber orientation ($91^N$). This is a very interesting and complex optimization problem. It also demonstrates how the laminated composite structure is sensitive to the change in fiber orientation. For instance, the exact critical pressure for the two-layered pressure vessel is $P_{cr}$=48.14 MPa (50°/58°). Applying manufacturing tolerances $t_{up}$=13° and $t_{low}$=7°, the actual burst pressure is $P_{cr}$=28.48 MPa (66°/35°), which is 41% less. This problem is demonstrated in Figure 9 where the four tolerance surfaces are shown. If we specify the nominal fiber orientation and during fabrication tolerances are incurred, the worst case scenario would result in $P_{cr}$=18.31 MPa, which is 64% less than the actual.

With the increase in the number of variables (layers), the impact of manufacturing tolerances becomes less severe but is still high enough to reckon with. For example, in the case of five layers, this difference drops to 36.16%. The calculations show that after 10 layers, there is not much improvement in the manufacturing uncertainty.

This example clearly demonstrates how important it is to take the manufacturing tolerances into account in the design optimization stage. It also shows the advantages and achievements in the application of modern algorithms.

## 5 Concluding remarks

The present study demonstrates how progress in modern evolutionary algorithms has revolutionized the design optimization of composite structures. The performance

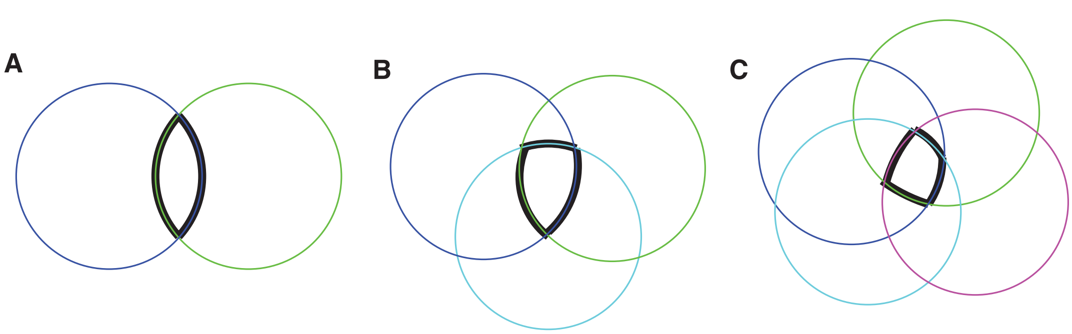


**Figure 8:** Schematic demonstration of the solution line in a 2-D design problem for (A) two; (B) three; (C) four tolerance surfaces.

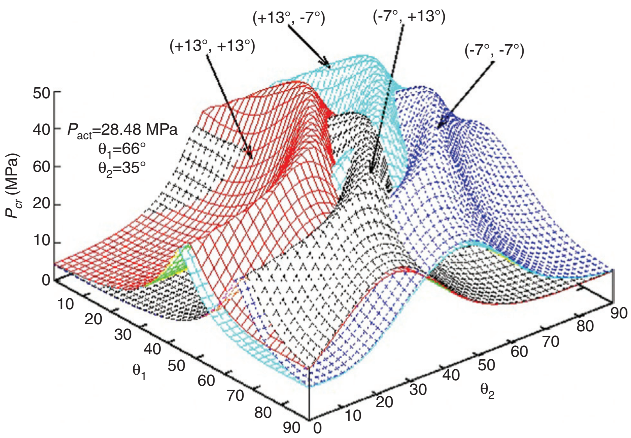


**Figure 9:** Effect of manufacturing tolerances on the *burst* pressure of the two-layered pressure vessel.

of such algorithms is shown by the example of the fiber-reinforced composite-laminated pressure vessel. It can be seen that with proper tuning such algorithms like PSO and the BB-BC optimization can reach the optimum solution within a few seconds even in a highly complex ten- and twenty-dimensional search space. It is obvious that similar results can be achieved for various other types of problems like weight and displacement minimization, shape optimization, analysis of manufacturing tolerances and many others.

It is, however, noted that the use of evolutionary algorithms does not always result in efficient optimization. For instance, for particular types of problems, the use of genetic algorithms might require millions of possible designs to be analyzed. Besides this, the heterogeneous material and geometrical complexities can stand in the way of an efficient search in a large design space. An accurate representation of the design model would lead to a much longer chromosome string and, as a result, a poor exchange of the genetic material and the stagnation of the algorithm. However, the genetic algorithm may succeed against all odds in optimizing almost any type of problems as it does not require its design parameters to contain explicit coordinate information.

At the same time, such methods like the PSO and the BB-BC algorithm seem to be faster and much easier to use than the GA if the design parameters can be given in the form of the coordinate numbers. This requirement, however, prevents their use for topological and similar design problems.

We expect even more difficulty being encountered in real life problems, and hence, we must have reliable multiobjective evolutionary algorithms (MOEA). It is noted that the stopping criteria is a weak point in MOEA, where setting anticipatorily the cutting point can be a specifically difficult task. The problem is that each separate population results in a single tradeoff during the optimal Pareto front approximation [22, 23]. Hybridization (combination) of such algorithms might significantly improve the performance of MOEA. Although much progress is already achieved, high dimensions of a search space will remain a formidable obstacle for an effective optimization because of the high density of the data and, as a result, loss of accuracy and somewhat "chaotic" behavior of the objective function.

## References

[1] Ganguli R. *J. Indian Institute Sci.* 2013, 93, 4.
[2] Holland JH. *Adaptation in Natural and Artificial Systems*, The University of Michigan Press: Ann Arbor, MI, 1976.
[3] Goldberg DE. *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley: Reading, MA, 1989.
[4] Mitsuo G, Runwei C. *Genetic Algorithms and Engineering Design*, Wiley: New York, 1997.
[5] Kennedy J, Eberhart R. *IEEE: 0-7803-2768-3/95/*. 1995, 4, 1942–1948.
[6] Erol OK, Eksin IA. *Adv. Eng. Softw.* 2006, 37, 106–111.
[7] Kripka M, Kripka RML. In *Proceedings of the International Conference on Engineering Optimization, EngOpt 2008*, Rio de Janeiro, Brazil, 01–05 June, 2008.
[8] Camp CV. *J. Struct. Eng*. 2007, 133, 999–1008.
[9] Kaveh A, Talatahari S. *Comput. Struct.* 2009, 87, 1129–1140.
[10] Gürdal Z, Haftka RT, Hajela P. *Design and Optimization of Laminated Composite Materials*, Wiley–Interscience: New York, USA, 1999.
[11] Rajeev S, Krishnamoorthy C. *J. Struct. Eng*. 1992, 118, 1233–1250.
[12] Le Riche R, Haftka RT. *AIAA J*. 1993, 31, 951–956.
[13] Lin C.-C. *Compos. Struct*. 2004, 63, 339–345.
[14] Omkar SN, Mudigere D, Narayana Naik G, Gopalakrishnan S. *Comput. Struct*. 2008, 86, 1–14.
[15] Parsopoulos KE, Tasoulis DE, Vrahatis MN. In *Proceedings of International Conference on Artificial Intelligence and Applications (IASTED)*, 2004.
[16] Tabakov PY. *Sci. J.: Bulletin of St. Petersburg State University of Engineering and Economics* 2011, 51, 150–172.
[17] Tabakov PY. In *Proceedings of International Conference on Computer, Electrical, and System Science, and Engineering, ICCESSE 2011*, Paris, France, 24–26 June 2011, pp. 1309–1313.
[18] Tabakov PY, Walker M. *Eng. Optim.* 2007, 39, 1–15.
[19] Tsai SW, Wu EM. *J. Compos. Mater*. 1971, 5, 58–80.
[20] Lekhnitskii SG. *Theory of Elasticity of An Anisotropic Elastic Body*, Holden-day, Inc.: San Francisco, 1963. Translated by P. Fern.
[21] Tabakov PY, Summers EB. *Comput. Struct*. 2006, 84, 374–384.
[22] Guerrero JL, Marti L, Berlanga A, Garcia J, Molina JM. In *IEEE Congress on Evolutionary Computation (CEC 2010)*, Barcelona, Spain, 18–23 July, 2010, pp. 4314–4321.
[23] Zitzler E, Laumanns M, Thiele L, Fonseca CM, Grunert da Fonseca V. In *GECCO'2002,* Morgan Kaufmann Publishers: San Francisco, USA, 2002, pp. 666–673.