Mugla Journal of Science and Technology

THE PERFORMANCE COMPARISON OF THREE METAHEURISTIC ALGORITHMS ON THE SIZE, LAYOUT AND TOPOLOGY OPTIMIZATION OF TRUSS STRUCTURES

Ali MORTAZAVİ*, Ege University, Graduate School of Natural and applied Sciences, Izmir, Turkey, ali.mortazavi.phd@gmail.com
(https://orcid.org/0000-0002-6089-7046)

Received: 18.07.2019, Accepted: 23.08.2019 Research Article
*Corresponding author DOI: 10.22531/muglajsci.593482

Abstract

The structural optimization problem mostly deals with the weight minimization of the structural system. This issue can be assessed from the size, layout and topology aspects. No matter which aspect(s) is targeted, to solve the problem an optimization technique is required. In the last decades the metaheuristic techniques, as the non-gradient optimization algorithms, are widely applied on solving these classes of problems. In the structural optimization, the most time consuming part of the process is the objective function evaluation. Based on this fact, in the current work, metaheuristic techniques are divided into three main groups as single phase, double phase and multi-phase algorithms. Then based on the author knowledge, three representative methods are picked for each group and their search performance comparatively inspected on solving size, shape and topology optimization of truss structures. To meet this aim, Integrated Particle Swarm Optimization (iPSO), Teaching and Learning Based Optimization (TLBO) and Drosophila Food-Search Optimization (DSO) algorithms are selected, respectively. Different properties like accuracy, convergence rate and complexity of the algorithms are investigated. The outcomes are provided via illustrative diagrams and tables. Based on the achieved results, DSO shows the most complexity level among the other algorithm while the iPSO and TLBO can outperform it on both accuracy and convergence rate. Consequently, iPSO presents a higher accuracy level on finding optimal solutions and TLBO with the lowest standard deviation value through the process shows the highest level of stability on finding optimal solutions.

Keywords: Structural Optimization, Metaheuristic Algorithms, Constrained Optimization, Performance Comparison

ÜÇ SEZGİSEL YÖNTEMİN KAFES SİSTEMLERİN TOPOLOJİ, GEOMETRİ VE BOYUT OPTİMİZASYONU ÜZERİNDE PERFORMANS KARŞILAŞTIRMASI

Özet

Bir yapısal optimizasyonda elemanların topolojisi, geometrisi veya kesitlerin boyutları dikkate alınarak sistemin ağırlığının minimize edilmesi amaçlanmaktadır. Çözüm tekniği olarak bu alanda son yıllarda üzerinde oldukça sık çalışılan sezgisel (metaheuristic) yöntemler geliştirilmiş ve kullanılmıştır. Yapısal optimizasyonda, amaç fonksiyonunun değerlendirmesi her iterasiyonda bir (ya da birden fazla) yapısal analiz gerektirmektedir ve dolaysıyla çözüm sürecinin en çok zaman alan kısmını oluşturmaktadır. Bu gerçeği dikkate alarak, mevcut çalışmada bu yöntemler, tek fazlı, çift fazlı ve çok fazlı algoritmalar olarak üç ana gruba ayrılmış ve her gruptan bir yöntem seçilmiştir. Daha sonra bu yöntemlerin arama performansları kafes yapıların boyut, geometri ve topoloji optimizasyonu üzerinde karşılaştırılmıştır. Entegre edilmiş Partikül Sürüsü Optimizasyon (EPSO), Öğretme ve Öğrenme esaslı Optimizasyon (ÖÖO) ve Derosofila Yiyecek arama Optimizasyon (DYO) sırasıyla seçilen algoritmalardır. Algoritmaların, yakınsama hızı, dikkati ve karmaşıklığı gibi farklı özellikleri değerlendirilmiştir. Elde edilen sonuçlara göre, DYO diğer algoritmalara kıyasen en yüksek karmaşıklık indeksine sahiptir, ayrıca EPSO ve ÖÖO dikkat ve yakınsama hızı açısından daha iyi performans göstermektedirler. Üstelik, EPSO, optimum çözümler bulma konusunda daha yüksek bir dikkat seviyesine sahiptir. Optimizasyon sürecinde ÖÖO en düşük standart sapma değerine sahiptir ve dolaysıyla optimum çözümler bulma konusunda en yüksek kararlılık seviyesini göstermektedir.

Anahtar Kelimeler: Yapısal Optimizasyon, Sezgisel Algoritmalar, Kısıtlı Optimizasyon, Performans Karşılaştırması

Cite

Mortazavi, A., (2019). "Verifying three types of metaheuristic algorithms on the size, layout and topology optimization of truss structures", Mugla Journal of Science and Technology, 5 (2), 28-41.

1 Introduction

The structural optimization generally deals with weight minimization of the structural systems. This aim can be met via considering different characteristics (e.g. size, shape and topology) of the desired structure. The size optimization tries to cover this aim via choosing the most proper cross-sectional areas for members of the system. The layout optimization deals with find the most proper nodal coordinates of the structural joints to reach on an optimal shape for the system. The topology optimization tries to find most proper configuration for the structural system via eliminating the unnecessary structural members. It is obvious that for the weight minimization problem, the objective function for all of them is the weight of the system.

The size optimization as the plainest approach can include either discrete or continuous search domain. However, for more realistic conditions the decision variables in this type of optimization problems should be selected from the discrete sets [1]. Generally, the number of probable unstable mechanisms in this kind of approaches is considerably low. However, the additional criteria (e.g. buckling criterion) on solving this class of problems can rise the complexity level of them [2]. Several investigations made in this field are available in the literature [3-5]. In the layout optimization of the structure, the nodal coordinates of the system are considered as the problem's decision variables [6-9]. Since the coordinate system is monolith, these cases generally deal with continuous search spaces. The difficulties in such approach can be emerge due to node proximities that causes to provide extreme low-length members with excessive rigidity that might negatively affect the analysis of the system. The topology optimization deals with finding most optimal configuration of the structural system via removing the unnecessary members from the system. One of the most complexities in handling such problems is to face excessive number of unstable mechanisms during the optimization process. Also, since the configuration of the structure is changed permanently the required coding system to handle existed varieties in the configuration is much more complex than two other approaches [10]. In comparison with other two approaches this class of optimization problems are less studied on the literature [10-13].

It is clear that for solving any optimization problem an optimization technique is required. The optimization techniques, generally, can be divided into two main groups: heuristic and non-heuristic approaches. The heuristic-based algorithms generally are rapid and accurate since they apply the continuous (or at least partially continuous) objective function and its gradients. But in several engineering optimization problem finding such objective function and/or the higher order gradients are very difficult or even impossible [14]. Subsequently, the metaheuristic algorithms can be employed as an alternative choice for such problems. These techniques mostly mathematically inspired from the natural events, physical rules or social behaviors. Metaheuristic algorithms are non-gradient based methods, so they don't require any continuous objective function and/or its higher gradients. Metaheuristic methods have population-based and iterative strategy in which while colony numerically search the domain each agent in each iteration tries to gradually improve its location [15]. One can chronologically list some of these methods as Differential Evolution (DE) [16], Ant Colony Optimization (ACO) [17], Hunting Search Algorithm (HsA) [18], Drosophila Food-Search Optimization (DSO) [19], Search Group Algorithm (SGA) [20], The Runner-Root Algorithm (RRA) [21] Quantum inspired Social Evolution (QSE) [22] and virus optimization algorithm (VOA) [23]. Each of cited algorithms has own affirmative and weak points in searching the different problem domains indeed their performance can be changed from case to case. These techniques as a global optimizer tools mostly are assessed on the non-constraint mathematical functions in their first emergence. In this type of problems objective function evaluation (OFEs) is just done by evaluating a certain function (e.g. polynomial or harmonic). So, the number of OFEs does not seriously affect the computational time. However, in the complex constrained engineering cases, like structural optimization problems, each objective function evaluation requires a structural analysis (e.g. finite element-based analysis) which is highly time consuming in comparison with calculation of fairly simple mathematical function. Thus, in this class of optimization problem the number of required OFE can be designated as the determinative factor on picking suitable method as the optimizer tool. So that, one method despite the presenting adequate performance on solving the simple mathematical cases can demonstrate unexpected search capability on the constrained structural problems. Accordingly, there are several comparative studies are performed on different techniques to assess them on the structural optimization field [24-27].

In the current study metaheuristic methods based on the number of objective function analyses (OFEs) performed in each iteration are divided into three main categories, those which perform single objective analysis (SOA) in each iteration, those which accomplish double objective analyses (DOA) in each iteration and those which make multiple objective analyses (MOA). Subsequently, three repetitive algorithms belong to SOA, DOA and MOA categories are selected (according to the author's knowledge) and their performances are verified on size, layout and topology optimization of the structural optimization problems.

2 Methods description

To provide and initial insight about the selected metaheuristic techniques this section is devoted to briefly explain the applied and assessed methods.

2.1 Teaching and Learning Based Optimizer (TLBO)

The teaching and learning based optimization introduced by Rao et al. [28] was inspired from the knowledge flow inside a classroom. The teacher educational influence on the students is considered as important factor in this method. Like the other natural-inspired algorithms, TLBO is a population based technique that begins with a set of arbitrary agents which are potential solution candidates. These random candidates are called learners. This algorithm is a two-phase technique included teaching and learning phases. The first phase concentrates on the transferring of knowledge from teacher to the learner(s) while second phase simulates the learning progression between the students through their pairwise communications. In the teaching phase all agents are assessed according to their objective function values then the best of them is chosen as the teacher. Afterward all agents adjust their positions considering the average knowledge grade of the classroom. If the updated situation is superior to prior location the new one is hold otherwise it is rejected. To perform leaning phase, an arbitrary pair of students is chosen and the agent (student) with lower knowledge level moves toward the more knowledgeable agent. Similarly, if the updated position is better than the prior one it is accepted and otherwise it is rejected. Connected with the given information TLBO algorithm is mathematically expressed as below:

$$\begin{aligned} &\mathbf{X}^{(new,i)} = \mathbf{X}^i + r(\mathbf{X}_{teacher} - T_F \mathbf{X}_{mean}) \\ &\text{if} \quad f\left(\mathbf{X}^{(new,i)}\right) < f(\mathbf{X}^i) \quad \mathbf{X}^i = \mathbf{X}^{(new,i)} \\ &\text{if} \quad f\left(\mathbf{X}^{(new,i)}\right) \geq f(\mathbf{X}^i) \quad \mathbf{X}^i = \mathbf{X}^i \end{aligned} \tag{1}$$

where, $\mathbf{X}^{(new,i)}$ is the updated location of i^{th} agent, $\mathbf{X}^i$ is its current position, T_F presents the teaching factor and can be either 1 or 2 also $f\left(\mathbf{X}^{(new,i)}\right)$ returns updated value of i^{th} agent while $f(\mathbf{X}^i)$ shows the current objective value of i^{th} agent.

Also, $\mathbf{X}_{mean}$ is the mean of all agents and it is defined as below:

$$\mathbf{X}_{mean} = \left[m\left(\sum_{j=1}^{np} x_j^1\right), m\left(\sum_{j=1}^{np} x_j^2\right), \dots, m\left(\sum_{j=1}^{np} x_j^{nd}\right) \right] \tag{2}$$

in which, np and nd indicate the number of the students and the problem dimension, respectively. $m(.)$ shows the mean value of any inputs. The learning phase mathematically is defined as below:

$$\begin{aligned} &\mathbf{X}^{(new,i)} = \mathbf{X}^i + r.\left(\mathbf{X}^i - \mathbf{X}^j\right) \quad if \quad f\left(\mathbf{X}^i\right) \leq f\left(\mathbf{X}^j\right) \\ &\mathbf{X}^{(new,i)} = \mathbf{X}^i + r.\left(\mathbf{X}^j - \mathbf{X}^i\right) \quad if \quad f\left(\mathbf{X}^i\right) > f\left(\mathbf{X}^j\right) \end{aligned} \tag{3}$$

Where, r is the random scalar and $\mathbf{X}^i$ and $\mathbf{X}^j$ are two independent members of the population. If $\mathbf{X}^{(new,i)}$ improves the objective prior value, it is accepted otherwise it is rejected and $\mathbf{X}^i$ is hold. For more clearness, the pseudo code for TLBO is demonstrated in Table 1:

Table 1. The pseudo code for TLBO.

Generating random agents (n agents) **while** (*termination conditions are not met*) *Selected the best agent as the teacher* **for** (each agent) *update each its location based on the teacher position via Eq. (2)* *evaluate updated(new) agent* $f(\mathbf{X}^i)$ **if** new location of the i^{th} agent is improved *maintain its new location* **else** *hold its prior location* **end**	*Teaching Phase*
select random j^{th} *agent while* $(i \neq j)$ **if** the of i^{th} agent is better than j^{th} agent i^{th} *agent getting far from the* j^{th} *agent using* Eq. (3) **else** i^{th} *agent moves toward the* j^{th} *agent based on* Eq. (3) **end** *evaluate updated(new) agent* $f(\mathbf{X}^i)$ **if** new location of i^{th} agent is improved *preserve its new location* **else** *reset it to its previous location* **end** **end** **end**	*Learning Phase*

2.2 Drosophila food-search optimizer (DSO)

The Drosophila food-Search optimizer (DSO) algorithm is the metaheuristic search method which mimics the food search treatment of the insect with the same name as Drosophila Melanogaster. This technique is population based method and it has been introduced for the first time by Das and Singh [19]. In this method two key paradigms are utilized to search the problem search space as Modified Quadratic Approximation (MQA) and neighborhood food searching. The neighborhood food searching pattern is formulated as follow:

$$\begin{aligned} &U_{i,k} = V_{i,k} + \left|V_{r3,k} - V_{r4,k}\right| \\ &W_{i,k} = V_{i,k} + \left|V_{r3,k} - V_{r4,k}\right| \; for \; k = r1 \text{ and } r2; \\ &\text{for } j \neq r1 \text{ and } j \neq r2, U_{i,k} = V_{i,j} \text{ and } W_{i,j} = V_{i,j} \\ &V'_{i,j} = \\ &\text{Min}\{f(V_{i,j}), f(U_{i,j}), f(W_{i,j})\} \text{ for } \begin{cases} i = 1,2,\dots,P \\ j = 1,2,\dots,D \end{cases} \end{aligned} \tag{4}$$

where, $i \in \{1,2,\dots,P\}$ which P is the population size, also $j \in \{1,2,\dots,D\}$ which D is the problem dimension. $r1$, $r2 \in [1, D]$ are tow random numbers. Also, $V_{i,k}$ is the current agent's location and $V'_{i,j}$ is the updated agent's position. The Modified Quadratic Approximation (MQA) search method is mathematically defined as below:

$$\text{Child} = 0.5\frac{(R_2^2-R_3^2)f(R_1)+(R_3^2-R_1^2)f(R_2)+(R_1^2-R_2^2)f(R_3)}{(R_2-R_3)f(R_1)+(R_3-R_1)f(R_2)+(R_1-R_2)f(R_3)} \quad (5)$$

in which $f(.)$ returns the value of objective function for any desired agent and R_1, R_2 and R_3 are selected arbitrarily from the colony so that $R_1 \neq R_2 \neq R_3$. For more directness, the pseudo code for DSO is given in Table 2.

Table 2. The pseudo code for DSO.

```
Initialize algorithm's internal parameters;
Evaluate objective function of each individual
while (not termination condition)
Apply tournament selection
  for (each particle i)
     Update the current agent making the neighborhood
     search using Eq. (4);
     Evaluate objective function for each agent f(Xi) using
     Eq. (4);
     The best agent is saved;
     If the fitness variation of any agent and its old position is
     within 1%, then providing MQA using Eq. (5);
     The new agent position will only maintain if it is better
     than its old position;
  end
end
```

2.3 Integrated particle swarm optimizer (iPSO)

The integrated particle swarm optimization (iPSO) algorithm is the PSO based algorithm which applies two new concepts as weighted particle to search the problem domain. To prevent from any misleading, it should be noted that iPSO has been introduced in 2016 by the author of the current work [10] and it differs from other method with the similar name (iPSO with small *i* latter is different from IPSO with capital I) in the literature. The proposed iPSO uses two different search patterns to navigate the particle toward the global optimum. One of these patterns is to move toward the global best found location ($\mathbf{X}^G$), previous best location stored in particles memory ($\mathbf{X}^P$) and weighted particle ($\mathbf{X}^w$). However, to meet the convergence criterion the algorithm performs another search pattern to navigate the particle by moving toward the gravity center of the colony which is determined by the weighted particle. This method mathematically is formulated as below:

if $\text{rand}_{0i} \leq \alpha$

$$^{t+1}\mathbf{v}_i = \varphi_{4i}(^t\mathbf{X}^w - {^t\mathbf{X}_i})$$

for $\varphi_{4i} = C_4 \times \text{rand}_{4i}$

if $\text{rand}_{0i} > \alpha$

$$^{t+1}\mathbf{v}_i = w_i \times {^t\mathbf{v}_i} + (\varphi_{1i}+\varphi_{2i}+\varphi_{3i})(^t\mathbf{X}_j^P - {^t\mathbf{X}_i}) + \varphi_{2i}(^t\mathbf{X}^G - {^t\mathbf{X}_j^P}) + \varphi_{3i}(^t\mathbf{X}^w - {^t\mathbf{X}_j^P}) \quad (6)$$

for $j \leq M$

$$\varphi_{1i} = C_1 \times \text{rand}_{1i}$$
$$\varphi_{2i} = C_2 \times \text{rand}_{2i}$$
$$\varphi_{3i} = C_3 \times \text{rand}_{3i}$$

and

$$^{t+1}\mathbf{X}_i = {^t\mathbf{X}_i} + {^{t+1}\mathbf{v}_i}$$

superscripts "*t*" and "*t*+1" indicate the current and updated values for a variable, respectively. $^{t+1}\mathbf{v}_i$ is the updated velocity, w_i is the inertia factor of prior movement of i^{th} particle. Also, the acceleration factors are described with $C_1 = -(\varphi_{2i}+\varphi_{3i})$, $C_2 = 2$, $C_3 = 1$, and $C_4 = 2$, and rand_{ki} where $k \in \{0,1,2,3,4\}$, is the random scalar designated from interval of [0, 1] [29]. This method applies the weighted particle ($\mathbf{X}^w$) is the gravity center of the colony and tries to collect the particles experiences according proportional with their fitness level. It mathematically is defined as follows:

$$\mathbf{X}^w = \sum_{i=1}^{M} \bar{c}_i^w \mathbf{X}_i^P$$
$$\bar{c}_i^w = \left(\hat{c}_i^w \Big/ \sum_{i=1}^{M} \hat{c}_i^w\right) \quad (7)$$
$$c_i^w = \frac{\underbrace{\max}_{1\le f\le M}\left(f(\mathbf{X}_{kw}^P)\right) - f(\mathbf{X}_i^P) + \varepsilon}{\underbrace{\max}_{1\le kw\le M}\left(f(\mathbf{X}_{kw}^P)\right) - \underbrace{\min}_{1\le kw\le M}\left(f(\mathbf{X}_{kw}^P)\right) + \varepsilon}, \quad i = 1,2,\ldots,M$$

in which M denotes the number of population of the colony. Proposed iPSO utilizes the Improved Fly-Back (IFB) technique to handle the problems' constraints [1]. However, to provide a fair comparative condition in the current wok it also utilizes the same penalty approach which the other methods are employed. This penalty approach is illustrated in the next section. For more clarity the pseudo code for iPSO is given in Table 3:

Table 3. The pseudo code for iPSO.

```
Initialize random particles
while (the termination criteria are not met)
  Calculate the weighed particle X^w using Eq.7
   for (each particle in the colony)
        if (rand_0i ≤ α)
          Calculate the velocity vector applying Eq.6
        elseif (rand_0i > α)
          Calculate the velocity vector applying Eq.6
        end
       Update the current particle
   end
  verify weighted particle condition
   if (f(X^w)) < (f(X^G))
     Set X^G = X^w,    (whenever X^w has the less objective
    function value replace with X^G)
   end
end
```

3 Handling the constraints

As mentioned in prior section, although iPSO applies the IFB for handling the problems' constraints, to provide more uniform comparative condition the penalty method is employed for all selected methods. Since the metaheuristic approaches generally are non-constrained methods, to handling the constraints of the problems in this investigation the penalty method is utilized. In this regard the penalty function is applied as below:

$$f_{penalty}(\mathbf{X}) = (1 + \varepsilon_1 v)^{\varepsilon_2} \times f(\mathbf{X})$$
$$v = \sum_{i=1}^{q} \max\{0, g_i(\boldsymbol{X})\} \tag{8}$$

in which, $f_{penalty}$ is the penalized objective function of the problem and $f(\mathbf{X})$ is the regular objective function value and $g_i(\boldsymbol{X})$ returns the ith problem constraint's violation. To provide more adaptive scenario, ε_1 and ε_2 as the tuning terms of the penalty function are respectively taken as 1 and 1.5 at the commence while linearly increased up to 6 [30].

4 Numerical examples

In this section the search performances of three different selected algorithms are comparatively assessed on three different classes of structural optimization problems. It should be noted that OFEs for structural analyses is reported more specific as number of structural analyses (NSAs) and they reflected the same concept. All computations are run on the computer equipped with the intel CORE i7@2.2 GHz CPU and 16 GB of RAM.

4.1 Sizing optimization of a spatial 582-bar tower

The first case is dedicated to the weight minimization of the spatial 582-bar tower shown in Figure 1 as a structural optimization example. In order to maintain the symmetry, the members of the structure are classified into 32 independent groups. There are three load conditions effective on the tower as:

I. The vertical load as -6.75 kips on each node
II. The horizontal load as 1.12 kips on each node in x-direction
III. The horizontal load as 1.12 kips on each node in y-direction

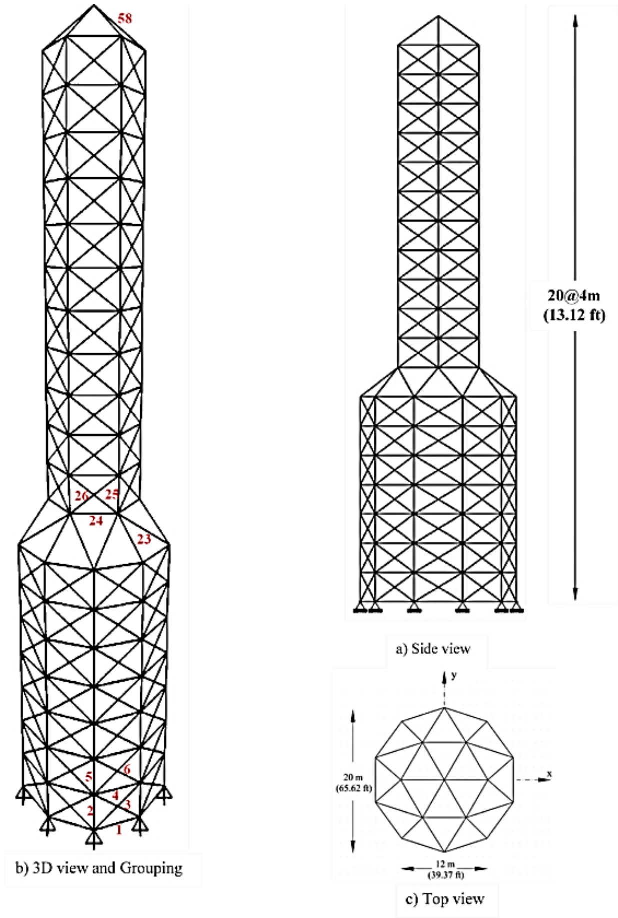

Figure 1. The 582-bar truss tower.

The sizing variables for this problem are selected from the discrete set given in Table 4. The members of this set contain 140 W-shape profiles given in steel structural profiles of AISC-ASD.

Table 4. W-shape profiles list taken from AISC code.

W27×178	W21×122	W18×50	W14×455	W14×74	W12×136	W10×77
W27×161	W21×111	W18×46	W14×426	W14×68	W12×120	W10×68
W27×146	W21×101	W18×40	W14×398	W14×61	W12×106	W10×60
W27×114	W21×93	W18×35	W14×370	W14×53	W12×96	W10×54
W27×102	W21×83	W16×100	W14×342	W14×48	W12×87	W10×49
W27×94	W21×73	W16×89	W14×311	W14×43	W12×79	W10×45
W27×84	W21×68	W16×77	W14×283	W14×38	W12×72	W10×39
W24×162	W21×62	W16×67	W14×257	W14×34	W12×65	W10×33
W24×146	W21×57	W16×57	W14×233	W14×30	W12×58	W10×30
W24×131	W21×50	W16×50	W14×211	W14×26	W12×53	W10×26
W24×117	W21×44	W16×45	W14×193	W14×22	W12×50	W10×22
W24×104	W18×119	W16×40	W14×176	W12×336	W12×45	W8×67
W24×94	W18×106	W16×36	W14×159	W12×305	W12×40	W8×58
W24×84	W18×97	W16×31	W14×145	W12×279	W12×35	W8×48
W24×76	W18×86	W16×26	W14×132	W12×252	W12×30	W8×40
W24×68	W18×76	W14×730	W14×120	W12×230	W12×26	W8×35
W24×62	W18×71	W14×665	W14×109	W12×210	W12×22	W8×31
W24×55	W18×65	W14×605	W14×99	W12×190	W10×112	W8×28
W21×147	W18×60	W14×550	W14×90	W12×170	W10×100	W8×24
W21×132	W18×55	W14×500	W14×82	W12×152	W10×88	W8×21

Their upper bound is limited to 6.16 in^2 (39.74 cm^2) and lower bound is limited to 215.00 in^2 (1387.09 cm^2). The nodal displacement for all main directions are limited up to 3.15 in. (8 cm). The stress limitation is calculated based on the buckling criterion of the AISD-ASD89 code as below [31]:

$$\begin{cases}\sigma_i^+ = 0.6\,F_y & \sigma_i \geq 0\\ \sigma_i^- & \sigma_i < 0\end{cases} \quad (9)$$

where F_y is the yielding stress of the materials and σ_i^- is compressive stress and σ_i^+ is tensile stress. While, σ_i^- is a function of the slenderness ratio given as below:

$$\sigma_i^- = \begin{cases}\left[\left(1-\frac{\lambda_i^2}{2C_c^2}\right)F_y \Big/ \left(\frac{5}{3}+\frac{3\lambda_i}{8C_c}-\frac{\lambda_i^3}{8C_c^3}\right)\right] & \text{for} \quad \lambda_i < C_c\\ \frac{12\pi^2 E}{23\lambda_i^2} & \text{for} \quad \lambda_i \geq C_c\end{cases} \quad (10)$$

in which C_c is the slenderness ratio which is described as:

$$C_c = \sqrt{\frac{2\pi^2 E}{F_y}} \quad (11)$$

According to the code, maximum slenderness ratio (i.e. allowable ratio) should be limited as up to 200 for compressive structural members and up to 300 for tensile structural members. The slenderness ratio is mathematically demonstrated as below:

$$\lambda_i = \frac{k_i l_i}{r_i} \leq \begin{cases}300 \text{ for tension members}\\ 200 \text{ for compression members}\end{cases} \quad (12)$$

where λ_i, r_i are slenderness ratio and radius of gyration of the i^{th} member, respectively. l_i indicates the length of the i^{th} member. If the required slenderness ratio is not be satisfied for compression elements, the allowable stress must not exceed the value of $\left(\frac{12\pi^2 E}{23\lambda_i^2}\right)$ ever [31]. This example as the complex sizing structural optimization is solved using selected methods and the achieved results are illustrated in Table 5. Standard deviation (Std.) and the number of objective function analyses (OFEs) for each algorithm are presented in this table. The convergence histories of the solution process for all approaches are comparatively plotted in Figure 2. As can be understood from these diagrams the algorithms can be sorted as iPSO, TLBO and DSO in terms of efficiency. The reached statistical data (i.e. shown via standard deviation) demonstrates that TLBO has the most stable condition on finding optimal solution among these three techniques. As the accuracy viewpoint iPSO outperforms the others by finding the lightest structure system.

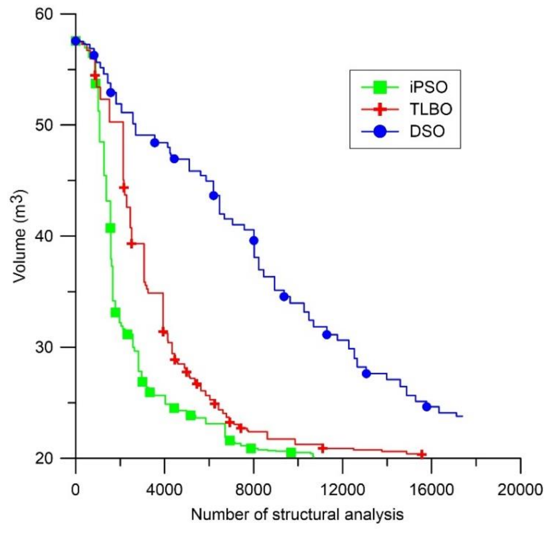

Figure 2. Convergence history for 582-bar tower size optimization.

Table 5. Comparison of the optimal results for the 582-bar tower problem.

Group	Optimal cross-sectional areas		
	TLBO	DSO	iPSO
1	W8×21	W8×24	W8×21
2	W24×84	W12×72	W24×76
3	W8×21	W8×28	W8×21
4	W24×62	W12×58	W12×65
5	W8×21	W8×24	W8×21
6	W8×21	W8×24	W8×21
7	W16×57	W10×49	W10×54
8	W8×21	W8×24	W8×21
9	W8×21	W8×24	W8×21
10	W12×53	W12×40	W12×50
11	W8×21	W12×30	W8×21
12	W10×77	W12×72	W10×68
13	W21×83	W18×76	W24×76
14	W21×57	W10×49	W14×53
15	W18×76	W14×82	W12×79
16	W8×21	W8×31	W8×21
17	W10×22	W14×61	W12×65
18	W18×55	W8×24	W8×21
19	W8×21	W8×21	W8×21
20	W8×21	W12×40	W12×45
21	W14×30	W8×24	W8×21
22	W8×21	W14×22	W8×21
23	W8×21	W8×31	W16×26
24	W8×21	W8×28	W8×21
25	W8×21	W8×21	W8×21
26	W8×21	W8×21	W8×21
27	W10×22	W8×24	W8×21
28	W8×21	W8×28	W8×21
29	W8×21	W16×36	W8×21
30	W8×31	W8×24	W8×21
31	W8×21	W8×21	W8×21
32	W12×22	W8×24	W8×21
Volume (m^3)	20.322	22.069	20.07
OFEs	16050	17670	12480
Std. (m^3)	0.22	0.51	0.24

As can be understood from attained results, iPSO can found the lightest structure. Also, considering the number of required objective function analyses the iPSO displays higher convergence rate. As the stability viewpoint TLBO by catching the lowest standard deviation exhibits the most stable behavior among all other techniques.

4.2 Size and layout optimization of a 52-bar dome under multiple frequencies constraint

The current example is devoted to optimizing the 52-bar dome structure shown in

Figure 3. In this example both size and shape properties of the structure are considered as the decision variables of the optimization process. Material density and modulus of elasticity are and 210 GPa 7800 kg/m^3, respectively. To hold symmetry of the system all members of the structure are put into 8 independent groups. Also free nodes (i.e. those are not restrained) are allowed to move ±2 m for their initial position shown in

Figure 3 in all principal directions, but again symmetry of system should be maintained. All free nodes are subjected to 50 kg non-structural mass. The first two natural frequencies are limited as $\omega_1 \geq 15.916$ Hz and $\omega_2 \geq 28.649$ Hz. Sizing variables should be selected between 1 and 10 cm^2 values. The achieved optimal results comparatively are tabulated in Table 6. Based on given outcomes iPSO can find most optimal solution, and TLBO and DSO stands fined nearest solution, respectively. Attained first four natural frequencies for this structure is presented in Table 7. For more clarity the found optimal layout for the structure is shown in Figure 4. Also, Figure 5 represents the convergence histories of selected three methods for the optimization process. As can be seen from this figure, iPSO, shows highest convergence rate among all other algorithms.

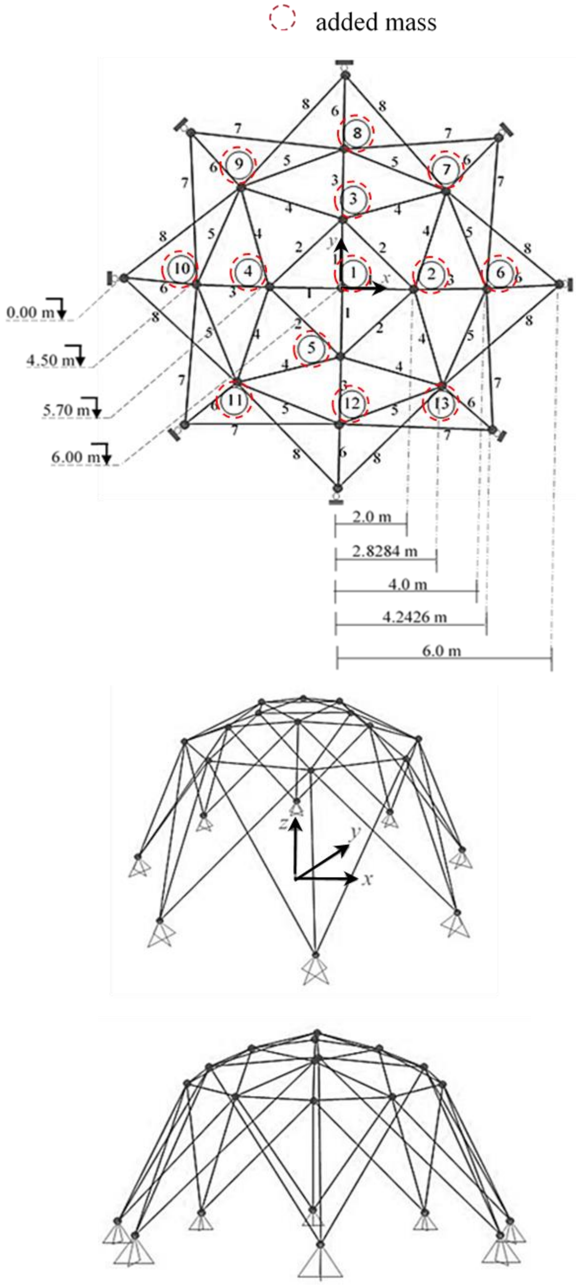

Figure 3. The 52-bar dome structure.

Table 6. Comparison of optimized designs obtained for the 52-bar dome truss structure.

Design Variables Z, X (m); A(cm²)	TLBO	DSO	iPSO
Z_1	5.9362	6.0188	6.0448
X_2	2.2416	2.2976	2.4276
Z_2	3.7309	3.7417	3.7076
X_6	3.963	3.9996	4.0391
Z_6	2.500	2.5001	2.5000
A_1- A_4	1.0001	1.0000	1.0000
A_5- A_8	1.1654	1.0852	1.0000
A_9- A_{16}	1.2323	1.1968	1.1840
A_{17}- A_{20}	1.4323	1.4503	1.4239
A_{21}- A_{28}	1.3901	1.4216	1.4339
A_{29}- A_{36}	1.0001	1.0001	1.0015
A_{37}- A_{44}	1.6024	1.5614	1.5665
A_{45}- A_{52}	1.4131	1.3878	1.3931
Best Weight (kg)	194.86	193.21	193.71
Mean Weight (kg)	196.95	199.42	200.8
Std.	2.38	3.99	6.98
NSA	12810	18210	8640

Table 7. Attained first four natural frequencies (Hz) of the 52-bar dome structure.

f (Hz)	TLBO	DSO	iPSO
1	12.8100	12.7510	11.4339
2	28.6500	28.6490	28.6480
3	28.6500	28.6490	28.6480
4	29.5400	28.8030	28.6482

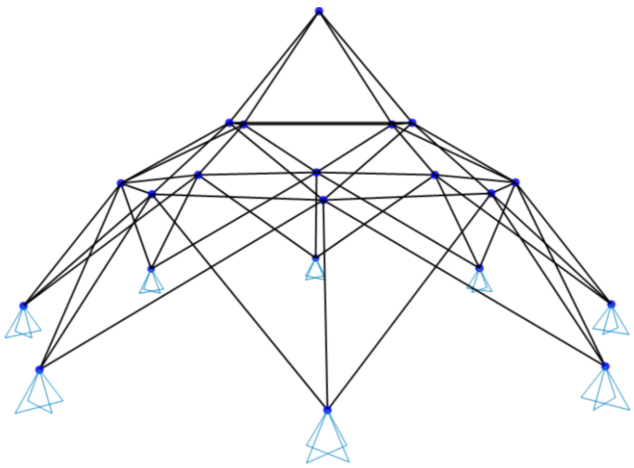

Figure 4. Optimal layout found for 52-bar dome.

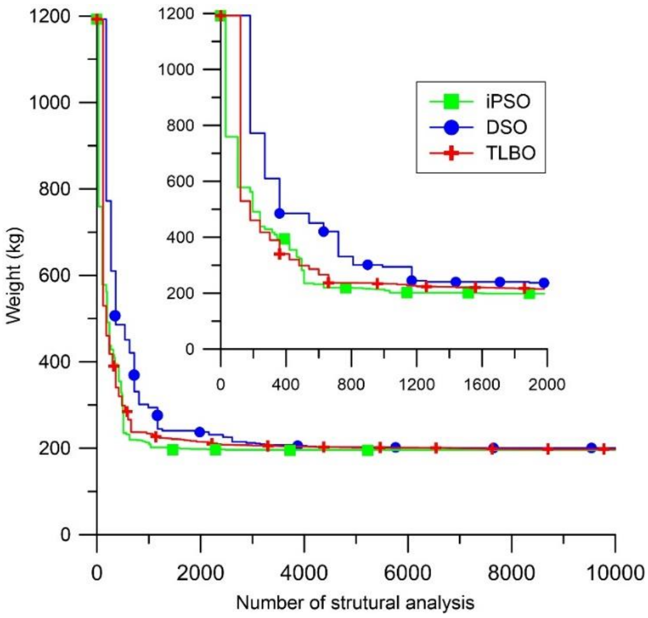

Figure 5. Convergence history for 52-bar dome size and layout optimization.

4.3 Size, layout and topology optimization of 47-bar transmission tower

As the final example size, layout and topology optimization of 47-bar transmission tower presented in Figure 6 is considered. Required boundary conditions and constraints of the problem are given in Table 8. Also, Table 9 addresses the load conditions adopted to structure system. To maintain structural symmetry, the members are categorized into 27 independent groups. The coordinates of nodes of 15, 16, 17 and 18 are fixed while just x-coordinate of nodes number 1 and 2 are permitted to be variable. Considering the number of design variables (17 shape and 27 size and topology variables) and kinds of them (size, layout and topology) this problem is the complex case. The achieved optimal structure is schematically represented in Figure 7. Also, obtained numeric results for all methods comparatively are tabulated in Table 10.

As can be seen from this table iPSO found lightest structural system while TLBO with the lowest standard deviation (Std.) shown the most stable behavior on finding the final solution. The convergence histories for all methods plotted in Figure 8. Based on these diagrams DSO has highest convergence rate among the tested methods.

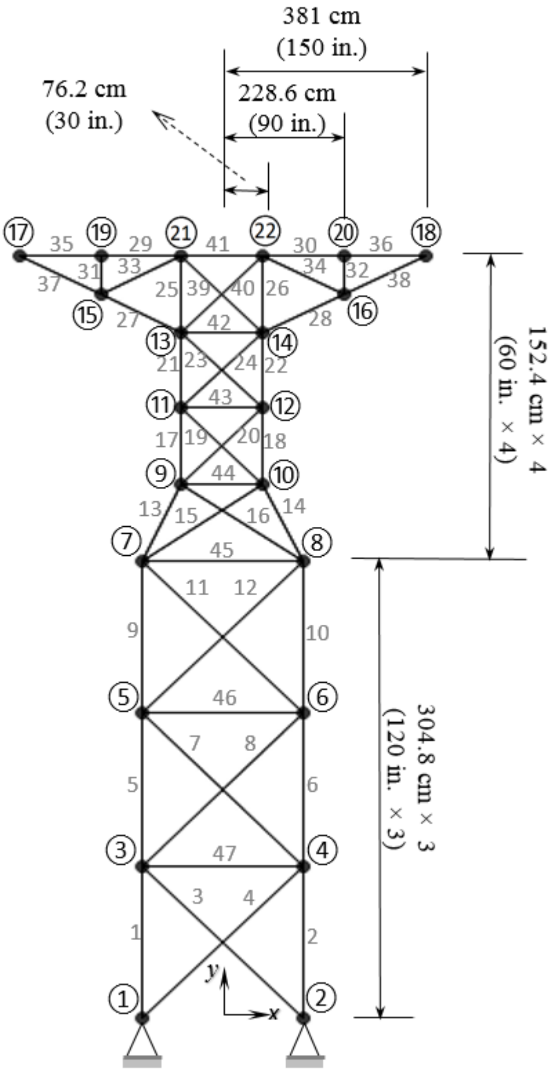

Figure 6. The ground structure for 47-bar transmission tower.

Table 8. Design variables constraints and constraints for 47-bar transmission tower.

Design variables	
Size and topology variables	Shape variables
A_m=A_{m-1} for m=2,4,6 , . . . , 40; A_{41}, A_{42}, A_{43}, A_{44}, A_{45}, A_{46}, A_{47}	x_2 =-x_1 ; x_4=-x_3 ; y_4=y_3 ; x_6=-x_5 ; y_6=y_5 ; x_8=-x_7 ; y_8=y_7 ;x_{10}=-x_9 ; y_{10}=y_9 ; x_{12}=-x_{11} ; y_{12}=y_{11} ; x_{14}=-x_{13} ; y_{14}=y_{13} ; x_{20}=-x_{19} ; y_{20}=y_{19} ; x_{22}=-x_{21} ; y_{22}=y_{21}
Shape variables	
0 ≤ x2, x4, x6, x8 ≤ 381.0 cm (150 in.) *0 ≤ x10, x12, x14 ≤ 228.6 cm (90 in.)* *0 ≤ x20 ≤ 381.0 cm (150 in.)* *0 ≤ x22 ≤ 228.6 cm (90 in.)* *0 ≤ y4 ≤ 609.6 cm (240 in.)* *304.8 cm (120 in.) ≤ y6 ≤ 914.4 cm (360 in.)* *609.6 cm (240 in.) ≤ y8 ≤ 1066.8 cm (420 in.)* *914.4 cm (360 in.) ≤ y10 ≤ 1219.2 cm (480 in.)* *1066.8 cm (420 in.) ≤ y12 ≤ 1371.6 cm (540 in.)* *1219.2 cm (480 in.) ≤ y14 ≤ 1524.0 cm (600 in.)* *1371.6 cm (540 in.) ≤ y20, y22 ≤ 1676.4 cm (660 in.)*	
Size variable set	
A_i Є [0.1,0.2,0.3,...,4.9,5.0], i = 1 , . . . , 47	
Constraint data	
Stress constraints	$(\sigma_i)_{ten}$ ≤ 137.8951 MPa (20 ksi), i = 1 , . . . , 47 $(\sigma_i)_{com}$ ≤ 103.4213 MPa (15 ksi), i = 1 , . . . , 47
Euler buckling constraints	$(\sigma_i)_{com}$ ≤ $-BEA_i / L_i^2$ i = 1 , . . . , 47
Material properties	
Modulus of elasticity, *E*	206842.7184 MPa (30000 ksi)
Buckling coefficient, *B*	3.96
Density, *ρ*	81.4341 kN/m3 (0.3 lb/in³)

Table 9. Load conditions for 47-bar transmission tower.

Loading condition	Joint	P_x(lbf)	P_y(lbf)
1	17, 18	26.6893 kN (6000)	-62.2751 kN (-14000)
2	17	26.6893 kN (6000)	-62.2751 kN (-14000)
3	18	26.6893 kN (6000)	-62.2751 kN (-14000)

Table 10. Optimum result for 47-bar transmission tower.

Design variables	TLBO	DSO	iPSO
A_1	3.2	3.3	3.3
A_3	1	0.3	0.3
A_5	3.0	2.9	2.9
A_7	1.1	1.4	1.3
A_9	3.0	3.0	3.1
A_{11}	0.5	0.6	0.5
A_{13}	2.7	2.6	2.6
A_{15}	1	1.1	1.2
A_{17}	2.6	2.9	2.8
A_{19}	0.8	0.5	0.5
A_{21}	2.4	2.4	2.3
A_{23}	1.1	1.7	1.7
A_{25}	0.6	0.5	0.5
A_{27}	1.7	1.6	1.5
A_{29}	1	0.8	0.8
A_{31}	1.1	1.1	1.2
A_{33}	0.5	0.8	0.7
A_{35}	1	1	1.1
A_{37}	1.4	1.3	1.4
A_{39}	1	0.9	0.9
A_{41}	0.8	0.5	0.4
A_{42}	Removed	Removed	Removed
A_{43}	Removed	Removed	Removed
A_{44}	Removed	Removed	Removed
A_{45}	Removed	Removed	Removed
A_{46}	Removed	Removed	Removed
A_{47}	Removed	Removed	Removed
X_2	103.111	112.0010	112.0
X_4	93.0012	88.1220	86.4461
Y_4	116.0015	140.1210	149.7223
X_6	72.0031	68.0009	68.3442
Y_6	222.3325	241.0001	241.5726
X_8	65.0087	61.0201	59.9921
Y_8	301.9981	326.0091	325.0389
X_{10}	50.0001	47.0011	47.8354
Y_{10}	390.1236	410.1009	407.9877
X_{12}	46.0112	44.0212	44.5377
Y_{12}	458.1122	450.1021	449.3806
X_{14}	51.0987	65.1229	63.3270
Y_{14}	507.0542	502.0091	500.6626
X_{20}	90.1009	58.0021	63.5867
Y_{20}	626.0092	635.0912	632.4550
X_{22}	19.9002	1.212	2.9293
Y_{22}	595.2134	598.0	599.1655
Weight, lb.	1782.17	1786.03	1780.99
Mean	1850.43	1890.01	1867.09
Std.	50.87	76.99	62.13
NSA	28480	36600	25600

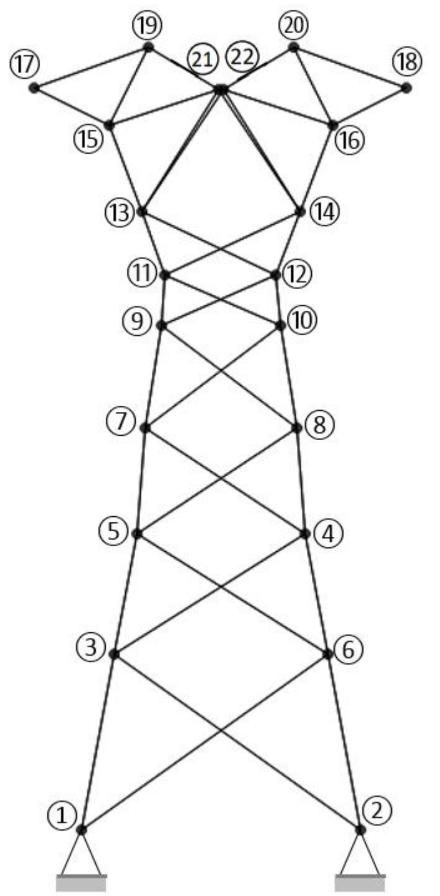

Figure 7. Obtained optimal structure for 47-bar transmission tower.

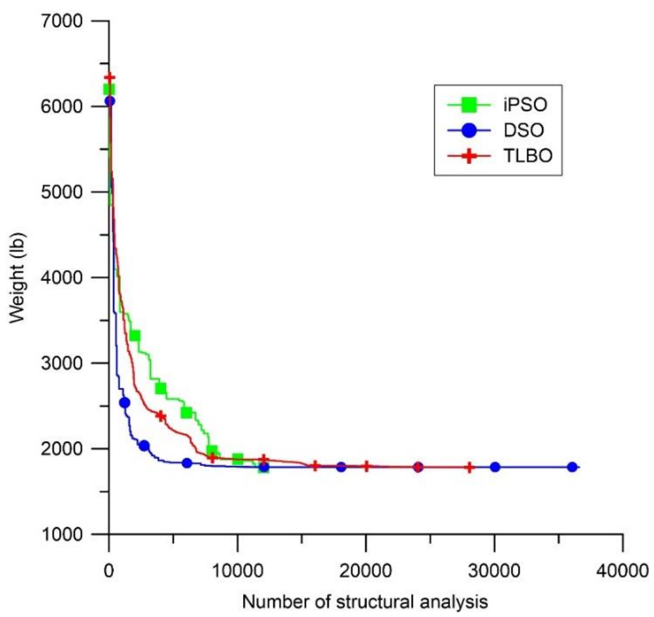

Figure 8. Convergence history for 47-bar dome size, layout and shape optimization.

5 Complexity analysis

This section is devoted to measuring the complexity level of the selected methods. To achieve this purpose, the suggested formulation in [32] is implemented. According to this formulation to determine an algorithm's complexity level three different times as *T0*, *T1* and $\hat{T}2$ should be evaluated as below [32]:

- *T0* should be calculated via 1000000 times calculating run time for following loop:

for *i*=1:1000000

x= 5.55 (*x* is double);

x=*x* + *x*; *x*=*x*./2; *x*=*x***x*; *x*=sqrt(*x*); *x*=ln(*x*); *x*=exp(*x*); *y*=*x*/*x*;

end

- *T1* is computed via 200000 times assessment of a designated function, where Griewank function is selected in current study. For specific dimensions as *D*=30 and *D*=50.

- *T2* is calculated via considering required time for the complete run of the proposed method over the same selected function and $\hat{T}2$ is the mean(average) value for five computation of *T2* times.

The complexity level for all methods are computed for Griewank function. To give more insight about this function it is schematically plotted in 3 dimensional form in Figure 9. All computations are executed on the same device and the results are presented in Table 11 and Table 12 for two different dimensions as *D*=30 and *D*=50 [32], respectively. As can be understood from provided data in these tables, TLBO algorithm is the most complex method while iPSO is the least complex technique among all methods. It is notable that in these tables the methods are ranked for complexity (e.g. the least complex method is ranked as 1). Also, The normalized complexity (i.e. normalized required time) for three structural optimizations is also represented in Table 13. As can be understood from the given data, DSO demands drastically higher computational time than two other methods, so it can be announced as the most complex algorithm among the selected approaches.

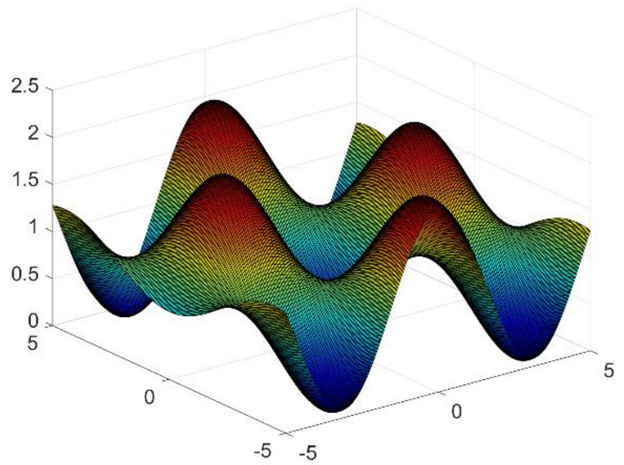

Figure 9. Griewank function.

Table 11. Complexity computation result for selected algorithms for D=30

Algorithm	$T0$	$T1$	$\hat{T}2$	$(\hat{T}2\text{-}T1)/T0$	Rank
TLBO	1.4E-01	1.9E-01	6.00E+02	4.28E+03	2
DSO	1.4E-01	1.9E-01	6.67E+02	4.76E+03	3
iPSO	1.4E-01	1.9E-01	4.60E+02	3.28E+03	1

Table 12. Complexity computation result for selected algorithms for D=50.

Algorithm	$T0$	$T1$	$\hat{T}2$	$(\hat{T}2\text{-}T1)/T0$	Rank
TLBO	1.40E-01	3.05E-01	5.59E+02	3.99E+03	2
DSO	1.40E-01	3.05E-01	7.94E+02	5.67E+03	3
iPSO	1.40E-01	3.05E-01	2.79E+02	1.99E+03	1

Table 13. Normalized complexity for selected structural problems.

Algorithm	[1]SO (582-bar)	[2]SLO (52-bar dome)	[3]SLTO (47-bar tower)
TLBO	2.3	2.8	2.9
DSO	3.5	3.6	4.1
iPSO	1	1	1

[1] SO: Size Optimization
[2] SLO: Size and Layout Optimization
[3] SLTO: Size, Layout and Topology Optimization

6 Conclusion

The current investigation deals with comparative assessments of three different optimization algorithms on different classes of the structural optimization problems. They are Teaching and Learning Based Optimization (TLBO), Drosophila Food-Search Optimization (DSO) and Integrated Particle Swarm Optimizer (iPSO) algorithms in the chronological order. From aspect of number of objective function evaluation (OFEs) these algorithms categorized as a single-phase, double-phase, and multi-phase algorithms. Indeed, in each iteration iPSO performs one OFE, TLBO makes two OFEs and DSO accomplishes more than two OFEs per each agent. The search capability of these methods are tested on three classes of structural optimization problems namely size optimization, simultaneously size and layout optimization and simultaneously size, layout and topology optimization of truss structures. Comparing achieved standard deviations reveals that TLBO with the lowest standard deviation value shows the most stable behavior on finding the optimal solution. Associating the convergence diagrams indicates that, except last benchmark structural problem iPSO has more rapid convergence rate compared with the other cited techniques. So that, in the last example nevertheless both TLBO and DSO converged fast, they stuck into local minima. Considering the point that in the structural optimization problems the number of NSAs plays prompted role on the efficiency and admissibility of the algorithm, the method with the lower number of required NSAs is highly more preferred. In this regard, iPSO by requiring the lower number of NSAs outperforms the other cited algorithms in the term of computational time. Also, performed complexity analyses reveals that single-phase iPSO is the least complex algorithm among all other addressed algorithms. Consequently, taking into account the achieved complexity index, accuracy level and required computational time, iPSO as the single-phase algorithm by establishing more proper performance is superior to the other examined algorithms on solving the structural optimization problems.

7 References

[1]Mortazavi, A., Toğan V., Nuhoğlu A., "An integrated particle swarm optimizer for optimization of truss structures with discrete variables", *Structural Engineering and Mechanics*, Vol. 61, pp. 359-370, 2017.
[2]Shon, S.-D., Hwang K.-J., Lee S.-J., "Numerical evaluation of buckling behavior in space structure considering geometrical parameters with joint rigidity", *Journal of Central South University*, Vol. 21 No. 3, pp. 1115-1124, 2014.
[3]Hasançebi, O., Kazemzadeh Azad S., "Discrete size optimization of steel trusses using a refined big bang–big crunch algorithm", *Engineering Optimization*, Vol. 46 No. 1, pp. 61-83, 2014.
[4]Hasançebi, O., Çarbaş S., Doğan E., Erdal F., Saka M.P., "Performance evaluation of metaheuristic search techniques in the optimum design of real size pin jointed structures", *Computers & Structures*, Vol. 87 No. 5, pp. 284-302, 2009.
[5]Chen, D., Zhao C., "Particle swarm optimization with adaptive population size and its application", *Applied Soft Computing*, Vol. 9 No. 1, pp. 39-48, 2009.
[6]Dede, T., Ayvaz Y., "Combined size and shape optimization of structures with a new meta-heuristic algorithm", *Applied Soft Computing*, Vol. 28, pp. 250-258, 2015.
[7]Mortazavi, A., Toğan V., "Sizing and layout design of truss structures under dynamic and static constraints with an integrated particle swarm optimization algorithm", *Applied Soft Computing*, Vol. 51, pp. 239-252, 2017.
[8]Gholizadeh, S., "Layout optimization of truss structures by hybridizing cellular automata and particle swarm optimization", *Computers & Structures*, Vol. 125, pp. 86-99, 2013.
[9]Hasançebi, O., Erbatur F., "Layout optimization of trusses using improved GA methodologies", *Acta Mechanica*, Vol. 146 No. 1, pp. 87-107, 2001.
[10]Mortazavi, A., Toğan V., "Simultaneous size, shape, and topology optimization of truss structures using integrated particle swarm optimizer", *Structural and Multidisciplinary Optimization*, Vol. 54 No. 4, pp. 715-736, 2016.
[11]Savsani, V.J., Tejani G.G., Patel V.K., "Truss topology optimization with static and dynamic constraints using modified subpopulation teaching–learning-based optimization", *Engineering Optimization*, Vol. 48 No. 11, pp. 17, 2016.
[12]Frans, R., Arfiadi Y., "Sizing, Shape, and Topology Optimizations of Roof Trusses Using Hybrid Genetic Algorithms", *Procedia Engineering*, Vol. 95, pp. 185-195, 2014.
[13]Rahami, H., Kaveh A., Gholipour Y., "Sizing, geometry and topology optimization of trusses via force method and genetic algorithm", *Engineering Structures*, Vol. 30 No. 9, pp. 2360-2369, 2008.
[14]Mortazavi, A., Toğan V., "Triangular units based method for simultaneous optimizations of planar trusses", *Advances in Computational Design*, Vol. 2 No. 3, pp. 195-210, 2017.
[15]Kennedy, J., Eberhart R., Particle swarm optimization, in: Neural Networks, 1995. Proceedings., IEEE International Conference on, 1995, pp. 1942-1948 vol.1944.

[16]Storn, R., Price K., "Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces", *Journal of Global Optimization*, Vol. 11 No. 4, pp. 341-359, 1997.

[17]Dorigo, M., Blum C., "Ant colony optimization theory: A survey", *Theoretical Computer Science*, Vol. 344 No. 2, pp. 243-278, 2005.

[18]Oftadeh, R., Mahjoob M.J., Shariatpanahi M., "A novel meta-heuristic optimization algorithm inspired by group hunting of animals: Hunting search", *Computers & Mathematics with Applications*, Vol. 60 No. 7, pp. 2087-2098, 2010.

[19]Das, K.N., Singh T.K., "Drosophila Food-Search Optimization", *Applied Mathematics and Computation*, Vol. 231, pp. 566-580, 2014.

[20]Gonçalves, M.S., Lopez R.H., Miguel L.F.F., "Search group algorithm: A new metaheuristic method for the optimization of truss structures", *Computers & Structures*, Vol. 153, pp. 165-184, 2015.

[21]Merrikh-Bayat, F., "The runner-root algorithm: A metaheuristic for solving unimodal and multimodal optimization problems inspired by runners and roots of plants in nature", *Applied Soft Computing*, Vol. 33, pp. 292-303, 2015.

[22]Pavithr, R.S., Gursaran. "Quantum Inspired Social Evolution (QSE) algorithm for 0-1 knapsack problem", *Swarm and Evolutionary Computation*, Vol. 29, pp. 33-46, 2016.

[23]Liang, Y.-C., Cuevas Juarez J.R., "A novel metaheuristic for continuous optimization problems: Virus optimization algorithm", *Engineering Optimization*, Vol. 48 No. 1, pp. 73-93, 2016.

[24]Kaveh, A., Zolghadr A., "Comparison of nine meta-heuristic algorithms for optimal design of truss structures with frequency constraints", *Advances in Engineering Software*, Vol. 76, pp. 9-30, 2014.

[25]Hasançebi, O., Çarbaş S., Doğan E., Erdal F., Saka M.P., "Comparison of non-deterministic search techniques in the optimum design of real size steel frames", *Computers & Structures*, Vol. 88 No. 17, pp. 1033-1048, 2010.

[26]Hammouche, K., Diaf M., Siarry P., "A comparative study of various meta-heuristic techniques applied to the multilevel thresholding problem", *Engineering Applications of Artificial Intelligence*, Vol. 23 No. 5, pp. 676-688, 2010.

[27]Mortazavi, A., Toğan V., Daloğlu A.T., Nuhoglu A., "Comparison of Two Metaheuristic Algorithms on Sizing and Topology Optimization of Trusses and Mathematical Functions", *Gazi University Journal of Science*, Vol. 31 No. 2, pp. 416-435, 2018.

[28]Rao, R.V., Savsani V.J., Vakharia D.P., "Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems", *Computer-Aided Design*, Vol. 43 No. 3, pp. 303-315, 2011.

[29]Mortazavi, A., Toğan V., Nuhoğlu A., "Weight minimization of truss structures with sizing and layout variables using integrated particle swarm optimizer AU - Mortazavi, Ali", *Journal of Civil Engineering and Management*, Vol. 23 No. 8, pp. 985-1001, 2017.

[30]Kaveh, A., Zolghadr A., "Democratic PSO for truss layout and size optimization with frequency constraints", *Computers & Structures*, Vol. 130, pp. 10-21, 2014.

[31]AISC. "American Institute of Steel Construction (AISC). Manual of steel construction allowable stress design. 9th ed. Chicago, IL", pp. 1-360, 1989.

[32]Suganthan, P.N., Hansen N., Liang J.J., Deb K., Chen Y., Auger A., Tiwari S., "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization", *Technical Report, Nanyang Technological University, Singapore, May 2005 AND KanGAL Report 2005005, IIT Kanpur, India*, 2005.