mühendislikdergisi
Dicle Üniversitesi Mühendislik Fakültesi

Dicle University Journal of Engineering (DUJE)

web: http://dergipark.gov.tr/dumf

Research Article

Comparative assessment of five metaheuristic methods on distinct problems

Ali Mortazavi [1,*]

[1] Ege University, Graduate School of Natural and applied Sciences, Ph.D, Izmir, Turkey, ORCID iD: 0000-0002-6089-7046

ARTICLE INFO

Article history:
Received 02.07.2019
Received in revised form 19 September 2019
Accepted 20 September 2019
Available online 26 September2019

Keywords:

Optimization techniques, Metaheuristic algorithms, Constrained optimization problems

Doi: 10.24012/dumf.585790

ABSTRACT

Metaheuristic algorithms belong to the non-gradient based optimization methods. Accomplished studies in this area reveal that each of these methods mostly has its own affirmative and inconvenient aspects. So that, one might provide a high level of exploration while the other can perform a great level of exploitation. Thus, selecting the proper and efficient algorithm for a problem can highly affect both the convergence rate and the accuracy level. There are several different metaheuristic algorithms have been announced in the technical literature in the last decade. Therefore, performing an objective comparative assessment over some of these methods can provide a fundamental and fair attitude for researchers either to select an algorithm which is more fitted with their target(s) or to develop even more efficient methods. So, the current investigation deals with evaluating and comparing of five different metaheuristic techniques emerged from ten years ago up to now. The selected methods can be sorted chronologically as Firefly Algorithm (FA), Teaching and Learning Based Algorithm (TLBO), Drosophila Food Search (DSO) method, Ions Motion Optimization (IMO) and Butterfly Optimization Algorithm (BOA). Different properties of these algorithms as convergence rate, diversity variation, complexity and accuracy level of the final solutions are compared on both constrained and non-constrained optimization problems include mathematical functions, mechanical and structural problems. The results show that the cited methods show different performance depending on the type of the optimization problem but overally BOA and TLBO outperform the other algorithms on non-constrained and constrained problems, respectively.

* Corresponding author
Ali, Mortazavi
✉ e-mail: ali.mortazavi.phd@gmail.com

Please cite this article in press as A. Mortazavi, "Comparative assessment of five metaheuristic methods on distinct problems", DUJE, vol. 10, no. 3, pp. 879-898, September 2019.

Introduction

In several science and engineering fields the main purpose of designing process is to establish the system with maximum efficiency and minimum cost [17-19]. Thus, the design problems somehow are convertible to an optimization problem and consequently optimization techniques play vital role on solving these problems. Optimization techniques can be divided into two main categories as deterministic and non-deterministic algorithms. Deterministic approaches indicate those methods that are based on mathematical modeling and programming techniques [7]. These approaches while initiate the search process from an initial design point by computing the gradient of the objective function explore the search space toward the optimum point. Despite the fact that such approaches have a fast convergence rate and high accuracy, they are highly dependent to the starting point and also they require a continues (or partial continues) objective function and its gradient(s). However, for many of engineering problems finding such an objective function is very difficult or even impossible [19].

So, to overcome these shortcomings another class of methods seems to be required. In this regard the non-deterministic approaches as an alternative optimization approach are emerged [10,24,25,30]. These class of optimization approaches do not entail gradient information of the problem's objective function and/or its constraints, if any [16,17,21]. They typically use probabilistic rules of transition rather than deterministic ones. They employ a number of randomly generated agents, and gradually improve them until the convergence/termination condition is met. Metaheuristic approaches, as main the techniques belong to this group, generally are inspired from natural phenomena. The basic clue behind these methods is to model natural concepts, like survival behavior of the animal colonies, physical rules and etc. [20]. Since these methods are numeric based, along with developments of the computers usage of these algorithm on different class of optimization problems are highly gained interest among the researchers [15,19].

It should be taken into account that there is no any unique manner or a standardized algorithm to implement of different metaheuristic approaches to solve different problems. Two important search behaviors which highly affect the search performance of the algorithm are exploration and exploitation of the algorithm. To spot the level of these behaviors and the ability of an algorithm to provide a suitable balance between these two different behaviors, the algorithm should be verified on different problems with various specifications. To meet this aim in the current study five well-stablished metaheuristic algorithms (based on the author's knowledge) are taken into consideration. It should be noted that, the investigated methods on this study are chosen to cover a recent decade innovations and achievements (from 2009 up to 2019) in the field of metaheuristic optimization techniques. These algorithms hierarchically can be sorted as Firefly Algorithm (FA) presented in 2009 by Yang [29], Teaching and Learning Based Optimization (TLBO) introduced in 2011 by Rao et al. [22], Drosophila Food Search (DSO) present in 2014 by Das and Singh [4], Ion Motion Algorithm (IMO) presented in 2015 by Javidy et al [8] and Butterfly Optimization Algorithm (BOA) presented in 2019 by Arora and Singh.

FA method imitating the firefly's mating behavior to search the domain and it used in several research works [11,14]. TLBO is modeled the educational relation between students and teacher in the classroom, this method and its modified versions also is utilizes as the optimizer tool in many different works [3,5,13,23,27,28]. DSO method models the food search strategies of the insect with the same name to search the problem domain [4]. This method uses two different patterns to globally and locally search the domain. IMO mathematically models the interactions between ions of the material in the liquefied and solid phases [6], and finally BOA method as the most recent method imitating the mating behavior of butterflies to provide a search algorithm.

Cited five algorithms' search capability are tested on variety of problems belonged to the mathematical, mechanical and structural fields. They are included both constrained and non-constrained search spaces with continuous and discrete variables. The methods different aspects like convergence rate and diversity index, complexity level and accuracy of the solution are verified on cited various classes of optimization problems and the results are provided via illustrative tables and comparative diagrams. The achieved results show that TLBO approach overlay shows a steady performance on searching the domain of different problems, while BOA demonstrates significant converging ability on non-constrained problem.

Optimization algorithms

This section is devoted to describe the methods that their search performances are investigated. All of these methods are metaheuristic algorithms. They are all population based algorithms which are started from an initial random agent while each agent is the candidate solution. These agents are gradually improved based problem according to the predefined pattern of the considered optimization algorithm. Assessment of the agented are done iteratively via evaluating the given objective function of the problem. Consequently, the applied methods in this study are briefly explained in the following.

Firefly Algorithm (FA)

The firefly algorithm (FA) is originally introduced by Yang [29], this method drives its search patterns form the behavior of the fireflies in the real world. It is inspired from the flashing behavior of fireflies and its absorbing effect on their own species. Although the real behavior of fireflies might be complex, to get a mathematical model the idealizations are made as follows:

- All fireflies are considered as unisex species; it means that all of them regardless to their sexuality they can be attracted to each other.
- The attractiveness factor directly proportional to their brightness. Therefore, for any pair of fireflies the brighter is more attractive and consequently the less bright firefly will move toward it. Also, this attractiveness depends on the distance between them, so it is decreased when the distance of two fireflies is increased and vice versa. On the other words if a firefly stands so far from others, it is not attracted by any of them and so it performs a random movement.
- The landscape of the objective function of the optimization highly affects the brightness of the fireflies (e.g. in the foggy weather even close fireflies may not realize each other).

Based on this information the two important terms in the firefly algorithm are light intensity of the agents and proper formulation to devote the attraction level of the agents. Since attraction of other fireflies to the light intensity of current firefly is depended to their distance, it can be defined via exponential function as below:

$$\beta(r_{ij}) = \beta_0 e^{-\gamma r_{ij}^2} \quad (1)$$

where, r_{ij} indicate the distance between a pair of fireflies and can be defined as Euclidean distance as $r_{ij} = \|\boldsymbol{X}_i - \boldsymbol{X}_j\|$ and β0 and γ indicate the attractiveness level for the r_{ij}=0 state and light absorption coefficient, respectively. Consequently, based on the given information the movement of the fireflies is mathematically formulated as follow:

$$\Delta\mathbf{x}_i = \beta_0 e^{-\gamma r_{ij}^2}(x_i - x_j) + \alpha\left(rand - \frac{1}{2}\right) \quad (2)$$
$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \Delta\mathbf{x}_i$$

where, superscript t+1 and t show the updated location of the firefly, respectively. Also, α is random number uniformly generated from [0,1] interval and 'rand' provides a vector of random number selected from [0,1] interval and β0=1 is set [29]. Finally, to provide more clarity the pseudo code for FA is given as below:

Table 1. The pseudo code for FA

Initialize internal algorithm parameters;
Generate n random firefly;
Light intensity I_i at $\mathbf{X}_i$ is determined by f($\mathbf{X}_i$)
while *(not termination condition)*
for *(each agent i)*
for *(each agent j)*
if *($I_j > I_i$), firefly i should move towards firefly j*
end
Adjust attractiveness according to distance r via exp[-γr]
Evaluate new solutions and update light intensity
end
end
end

Teaching and learning based optimization algorithm (TLBO)

The teaching and learning based optimization inspired from the knowledge flow inside a classroom was presented by Rao et al. [22]. This method considers the teacher educational influence on the students. Similar to other natural-inspired algorithms TLBO is a population based method that starts with a random class which contains possible solution candidates. These random candidates are called learners. This algorithm is two phase method which includes teaching phase and learning phase. The first phase focuses on the knowledge transferring from teacher to the learners while second phase models the learning process among the student through their individual interactions.

In the teaching phase all agents are evaluated based on their objective function values and the best agent is selected as the teacher. Then all agents are modified their locations based on mean knowledge level of the classroom. If the updated solution is better than prior one the new one is accepted and otherwise it is rejected. Subsequently, in the learning phase a random pair of students is selected and the student (agent) with lower level of knowledge moves toward the other with higher level of knowledge. Again if updated location is better than the previous one it is accepted and else it is rejected. Based the given information TLBO method is mathematically formulated as follow:

$$\begin{aligned} &\mathbf{X}^{(new,i)} = \mathbf{X}^{i} + r(\mathbf{X}_{teacher} - T_F\mathbf{X}_{mean}) \\ &if \quad \mathrm{f}(\mathbf{X}^{(\mathrm{new,i})}) < \mathrm{f}(\mathbf{X}^{\mathrm{i}}) \quad \mathbf{X}^{\mathrm{i}} = \mathbf{X}^{(\mathrm{new,i})} \\ &\text{if} \quad f(\mathbf{X}^{(new,i)}) \geq f(\mathbf{X}^{i}) \quad \mathbf{X}^{i} = \mathbf{X}^{i} \end{aligned} \tag{3}$$

where, $\mathbf{X}^{(new,i)}$ is the updated position of i^{th} agent, $\mathbf{X}^{i}$is its current location, the teaching factor is shown with TF and can be either 1 or 2 also $f(\mathbf{X}^{(new,i)})$ and $f(\mathbf{X}^{i})$ show updated and the current objective values of ith agent, respectively.

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

$$\mathbf{X}_{mean} = \left[m\left(\sum_{j=1}^{np} x_j^1\right), m\left(\sum_{j=1}^{np} x_j^2\right), \dots, m\left(\sum_{j=1}^{np} x_j^{nd}\right) \right] \tag{4}$$

in which, *np* shows the number of the students, *m*(.) returns the mean value of any inputs and *nd* indicates the problem dimension. The learning phase mathematically is formulated as follow:

$$\begin{aligned} \mathbf{X}^{(new,i)} &= \mathbf{X}^{i} + r.(\mathbf{X}^{i} - \mathbf{X}^{j}) \quad if \quad f(\mathbf{X}^{i}) \leq f(\mathbf{X}^{j}) \\ \mathbf{X}^{(new,i)} &= \mathbf{X}^{i} + r.(\mathbf{X}^{j} - \mathbf{X}^{i}) \quad if \quad f(\mathbf{X}^{i}) > f(\mathbf{X}^{j}) \end{aligned} \tag{5}$$

Where, r is the random value and $\mathbf{X}_i$ and $\mathbf{X}_j$ are two different member of the population. If $\mathbf{X}^{(new,i)}$ improves the objective value, it is accepted otherwise it is rejected and $\mathbf{X}_i$ is maintained. For more clarity, the pseudo code for TLBO is provided as follow:

Table 2. The pseudo code for TLBO

Generating n random agents **while** *(not termination condition)*	
Sort the colony and selected the best agent as teacher **for** *(each agent i)* *update each agents location considering teacher position via Eq. (4)* *evaluate updated agent (f(**X**i))* **if** *new location of agent i is improved* *maintain new location* **else** *reset it to its prior best location* **end**	*Teaching Phase*
select random j^th agent which (i≠j) **if** *new location of agent i is better than agent j location* *agent i getting away from the agent j based on Eq.(5)* **else** *agent i going toward from agent j based on Eq.(5)* **end** *evaluate updated agent (f(**X**i))* **if** *new location of agent i is improved* *maintain new location* **else** *reset it to its prior best location* **end** **end** **end**	*Learning Phase*

Drosophila search algorithm (DSO)

The Drosophila Search Algorithm (DSO) is the metaheuristic search approach which imitates the food search behavior of the insect with the same name as Drosophila Melanogaster. The method is population based approach and it was released for the first time by Das and Singh [4]. In this approach two main strategies are applied to search the problem domain as neighborhood food searching and Modified Quadratic Approximation (MQA). The neighborhood food searching is formulated as below:

$$U_{i,k} = V_{i,k} + |V_{r3,k} - V_{r4,k}|$$
$$W_{i,k} = V_{i,k} + |V_{r3,k} - V_{r4,k}| \; for\; k = r1 \; and \; r2;$$
$$for\; j \neq r1 \; and \; j \neq r2,\; U_{i,k} = V_{i,j} \; and \; W_{i,j} = V_{i,j} \qquad (6)$$
$$V'_{i,j} = Min\{f(V_{i,j}), f(U_{i,j}), f(W_{i,j})\} \; for \begin{cases} i = 1, 2, \dots, P \\ j = 1, 2, \dots, D \end{cases}$$

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

$$Child = 0.5 \frac{(R_2^2 - R_3^2)f(R_1) + (R_3^2 - R_1^2)f(R_2) + (R_1^2 - R_2^2)f(R_3)}{(R_2 - R_3)f(R_1) + (R_3 - R_1)f(R_2) + (R_1 - R_2)f(R_3)} \qquad (7)$$

in which $f(.)$ indicates the objective function value for any indicidual and R_1, R_2 and R_3 are slected randomly among the colony so that $R_1 \neq R_2 \neq R_3$. For more clarity, the pseudo code for DSO is provided in Table 3.

Table 3. The pseudo code for DSO

```
Initialize internal algorithm parameters;
Evaluate fitness of each individual in the populations
while (not termination condition)
Use tournament selection
  for (each agent i)
     For each agent in the population make the neighborhood search using Eq. (6) and update it
     Evaluate objective function of each member f(Xi) using Eq. (6)
     The best place of agent is saved
     If the fitness of any agent and its old position is within 1% radius, then apply MQA using Eq. (7)
The old individual will only retain its position if it is better than the current individual
  end
end
```

Ions Motion Optimization (IMO) Algorithm

The Ions Motion Optimization (IMO) mimics the ions behavior in nature, indeed it is inspired from repulsion and attraction force between cations and anions. This algorithm has two different navigation schemes as liquid and solid to guidance the agents. It is considerable that the algorithm of IMO has not any random coefficient in its main phase (i.e. liquid phase). So, in the liquid phase all agents (ions) are move toward each other non-stochastically. Also, since all of initial population are divided into two main groups of ions as anions and cations the population size should be an even number. In the solid phase agents are navigated toward (or around) the best solution to provide exploitation. Based on the method's authors this phase preventing from local optima trappings [8]. Based on provided information the liquid and solid phases of IMO respectively are formulated as below:

Liquid phase

$$A_{ij} = A_{ij} + AF_{ij} \times \left(Cbest_j - A_{ij}\right)$$
$$C_{ij} = C_{ij} + CF_{ij} \times \left(Abest_j - C_{ij}\right)$$

Solid phase

$$
\begin{aligned}
&\textbf{if} \begin{pmatrix} CbestFit \geq \frac{CworstFit}{2} \\ and \\ AbestFit \geq \frac{AworstFit}{2} \end{pmatrix} \\
&\quad \textbf{if } rand(\,) > 0.5 \\
&\qquad A_i = A_i + \Phi_1 \times (Cbest - 1) \\
&\quad \textbf{else} \\
&\qquad A_i = A_i + \Phi_1 \times (Cbest) \\
&\quad \textbf{end if} \\
&\quad \textbf{if } rand(\,) > 0.5 \\
&\qquad C_i = C_i + \Phi_2 \times (Abest - 1) \\
&\quad \textbf{else} \\
&\qquad C_i = C_i + \Phi_2 \times (Abest) \\
&\quad \textbf{end if} \\
&\quad \textbf{if } rand(\,) < 0.05 \\
&\qquad Re - initialized\ A_i\ and\ C_i \\
&\quad \textbf{end if} \\
&\textbf{end if}
\end{aligned}
\qquad (8)
$$

where *Cbest* and *Abest* are shown the best cation and anion, respectively. Also, Φ is the random number uniformly selected form interval of [0,1]. *CbestFit* and *AbestFit* are indicated the fitness values for the best cation and anion, respectively. AF_{ij} and CF_{ij} are force coefficient between ions and they are defined as follows:

$$AF_{ij} = \frac{1}{1 + e^{-0.1/AD_{ij}}}$$
$$CF_{ij} = \frac{1}{1 + e^{-0.1/CD_{ij}}} \qquad (9)$$

In which, $AD_{ij} = \left|A_{ij} - Cbest_j\right|, CD_{ij} = \left|C_{ij} - Abest_j\right|$ are distance between i^{th} agent and best cation and best anion, respectively. for more clarification the pseudo code for IMO is given as follows:

Table 4. The pseudo code for IMO

Initialize internal algorithm parameters and random population;
while *(not termination condition)*
  Fitness evaluation of all agents
  Determine the best and worst Anions and Cations
  Determine force factor using Eq. (9)
  Update locations of ions based on the liquid phase as given in Eq. (8)
  if *(Mean fitness of worst ions are equal or smaller than the best ions)*
    Perform solid phase motion based on the Eq. (8)
  end
end

Butterfly Optimization Algorithm (BOA)

The butterfly optimization algorithm models the biological behavior of the butterflies to find food sources and mating. This method has three main phases as initializing, main iterations and finalizing [2]. The created model can be idealized as follows: (a) It is supposed the all butterflies are able to emit the fragrance which causes butterflies attract to each other, (b) each butterfly provides a stochastic movement toward the best butterfly (i.e. the butterfly emitted most intense fragrance), (c) landscape of the search domain affect the level of stimulus intensity of each butterfly. For global and local searches, BAO applies two similar but different strategies as given in following:

Global search

$$x_i^{t+1} = x_i^t + \left(r^2 \times g^* - x_i^t\right) \times f_i$$

Local search $\qquad (10)$

$$x_i^{t+1} = x_i^t + \left(r^2 \times x_j^t - x_k^t\right) \times f_i$$

where, *t*+1 and *t* are shown the current and updated condition for the related variable. Also, g^* indicates the best agent in the colony while x_j and x_k are two randomly selected agents from the colony. The coefficient of *r* is a random number uniformly selected from [0,1]. The fragrance factor is defined as below:

$$f = cI^a \qquad (11)$$

where, *f* is the magnitude of the fragrance and *I* is the intensity of the stimulus and a is the value that accounts the fluctuating absorption degree. Also, *a* coefficient can take values between [0,1] and *c* can be selected from interval of [0, ∞]. However, the based on presenters of BOA the proper values for both of these coefficients are selected in range of [0,1]. For more clarity the pseudo code for BOA are given as below:

Table 5. The pseudo code for BOA

```
Initialize internal algorithm parameters
Generate n random butterflies
Calculate F(Xi) to determine stimulus intensity Ii
Define sensor modality c, switch probability p and power exponent a
while (not termination condition)
   for each population (butterfly)
Using Eq. ( 11 ) calculate the fragrance for current butterfly
  end
Select the best agent
  for each butterfly in the colony
              Generate rand number r
             If r<p then
Moving toward the best solution using global search of Eq. ( 10 )
             else
Perform a random movement using local search of Eq. ( 10 )
             end
  end
end
```

Numeric problems

In this section the different perfections and deficiencies of the addressed five metaheuristic algorithms are comparatively tested on a number of numeric mathematical and mechanical problems. These problems are covered both constrained and non-constrained cases. It should be noted that the main aim of the current work is not to make a fine tuning on any of cited algorithms over different types of problems, but their original and well stablished forms are applied to provide a fair comparison. To illustrate more details about these algorithms the internal terms' values are given as follow:

Table 6. Parameters values for utilized algorithm

Algorithm	Parameter values
FA [29]	α=0.25, β=0.2, γ=1
TLBO [22]	TF = round[1 + rand(0, 1){2 - 1}]
DSO	-
IMO [8]	-
BOA [2]	p=0.8, a=0.1, c=0.01

It should be mentioned that the computer codes provided by their own authors for FA and BOA published over the MathWorks® are applied. All problems are solved via the system equipped with intel CORE i7@2GHz with 16GM of RAM installed. To prevent any premature convergence all cases are run for 30 times.

Non-constrained benchmark functions

This section is devoted to test the performance of the selected algorithms on the optimization of the non-constrained benchmark functions. In this regard five best well-known benchmark functions with different properties are selected. To provide more clarity these functions are schematically plotted in the 3D space in Table 7 and their properties are also given in this table. The first two function relatively has smooth search space which challenge the global search behavior of the tested algorithms in addition the sphere function is the multimodal and separable while the Schwefel 2.22 function is unimodal and non-separable [12]. The Ackley function is a transmitting function for testing the algorithms. Since it has smooth but with vast uniform area of search space, while it highly challenges the global search behavior of the algorithms, it also requires an admissible level of local search behavior. This function search space is multimodal and non-separable. The functions of Griewank and Rastrigin both have high number of local optima, and as can be seen from Table 7 specially the later one has very noisy search domain which makes the local search very difficult for the algorithms. The related formulation, the range and assumed dimensions for the selected functions are provided in Table 8. All algorithms run for 10*D (D=dimension of the problem) number of objective function evaluation, also the error level for each example are taken as 1E-8 [26]. The error level indicates the error level between achieved objective function and global optimum and it is applied as alternative termination criterion. Indeed, both the maximum number of iteration and maximum error level can be a termination criterion depending on whichever occurred first.

Convergence analysis

All algorithms performances are examined on solving the five addressed benchmark functions given in Table 7. The related convergence diagrams for all algorithms are presented in Figure 1. For the sphere function all tested method expect IMO can reach optimal condition. As the most capable methods both TLBO and BOA show nearly the same performance with 6840 OFEs they are followed by DSO and FA with13020 and 16440 OFEs, respectively. For the Schwefel 2.22 function BOA with 2130 OFEs outperform all other tested methods, and TLBO stands at the second place by 9000 OFEs and DSO can reach the optimal condition through 32010 of OFEs. Both FA and IMO are not able to satisfy the optimal condition for this example. For the Ackley function, BOA outperforms all other methods via 1980 OFEs and it is traced by TLBO with 8640 OFEs. However, other tested algorithms do not reach to the optimal conditions. For the Griewank function the successful methods can be sorted in ascending order as BOA, TLBO and DSO with 2130, 6180 and 11790 OFEs, respectively. This is despite the fact that IMO and FA cannot attained optimal condition. For the Rastrigin function BOA shows extreme performance among all other tested methods and it reaches to the optimal condition within just 1980 OFEs. After it TLBO with 24360 OFEs other three methods are not able to satisfy any optimal condition. Overally looking to the obtained convergences histories for all tested methods it can be seen that BOA considerably outperforms all other methods while TLBO stands in the second place on solving the non-constrained functions. Subsequently FA, DSO and IMO methods stand in the next places, respectively.

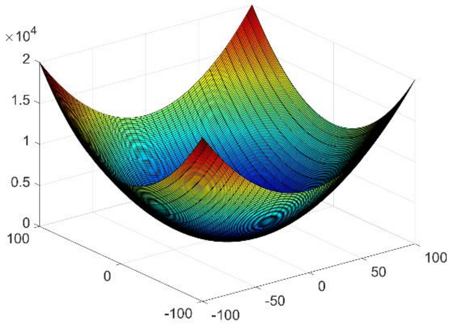
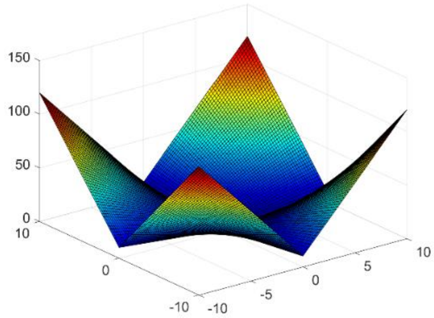
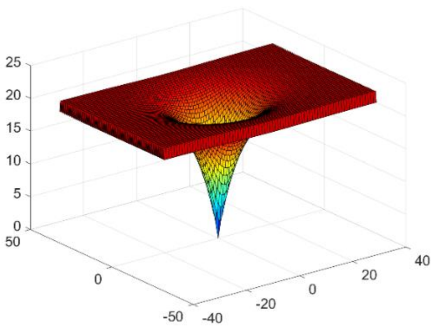
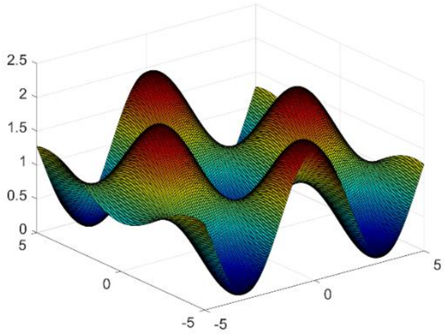
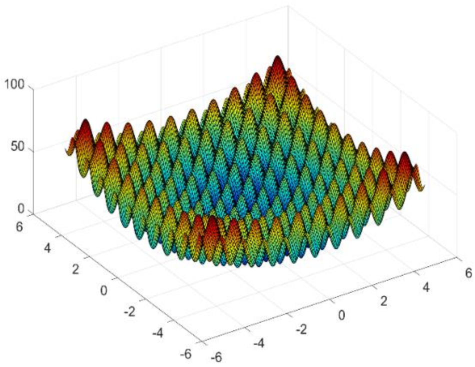
Diversity analysis

In the current section the diversity level of the selected five methods are assessed via providing the illustrative diversity history diagrams. All swarm based methods demand an admissible level of the diversity which provides both exploration and exploitation search behaviors. With lower level of diversity, the algorithm loses its ability on explorations search and can be easily trapped into local minima and premature converge seems to be unavoidable. However, extensive level of diversity makes the algorithm to fail on achieving the required convergence, since it cannot provide an efficient search in the proximity of the local optima. To illustrate the diversity level of the selected methods their diversity history is comparatively addressed. In this regard, to measure the diversity level the diversity index is applied as follow:

$$Diversity\ (t) = \frac{1}{N|L|}\sum_{i=1}^{N}\sqrt{\sum_{j=1}^{D}\left(x_i^j - \bar{x}^j\right)^2} \qquad (12)$$

where t designates the current step, number of population is shown by N, L indicates the longest diagonal length of the search space, D stands for the problem dimension, x_i^j is the j^{th} component of the i^{th} agent, and $\bar{x}^j$ declares the mean of the all j^{th} components for whole swarm. The corresponding diversity history diagrams of selected methods for two dimensional sphere function are given in Figure 2. As can be seen from this figure the diversity in TLBO and DSO are rapidly decreased and it means that all agents are strongly conducted toward the optimal point. Diversity level of BOA is decreased in lower rate and it seems this method can provide a mediocre level of diversity, since the convergence rate of this method is a way higher than the others. Consequently, the FA and IMO show fluctuated diversity level and if the smooth domain of the 2D sphere is considered, such a variation in the diversity is not required and these two methods don't perform an adaptive search behavior.

Table 7. Benchmark functions' schematic presentations and their specifications

	Specification*		Specification*
 Sphere	M,S	 Schwefel 2.22	U,N
 Ackley	M,N	 Griewank	M,N
 Rastrigin			M,S

* M: Multimodal, N: Non-Separable, U: Unimodal, S: Seperable

Table 8. Benchmark functions' definitions

Function	Specification	Range	Dimension
Sphere	$f(x) = \sum_{i=1}^{n} x_i^2$	[-100,+100]	30
Schwefel 2.22	$f(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	[-10,+10]	30
Ackley	$f(x) = -20 \exp\left(-0.2 \times \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n} \cos(2\pi x_i)\right) + 20 + \exp(i)$	[-32,32]	30
Griewank	$f(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[-600,+600]	30
Rastrigin	$f(x) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10)$	[-5.12,+5.12]	30

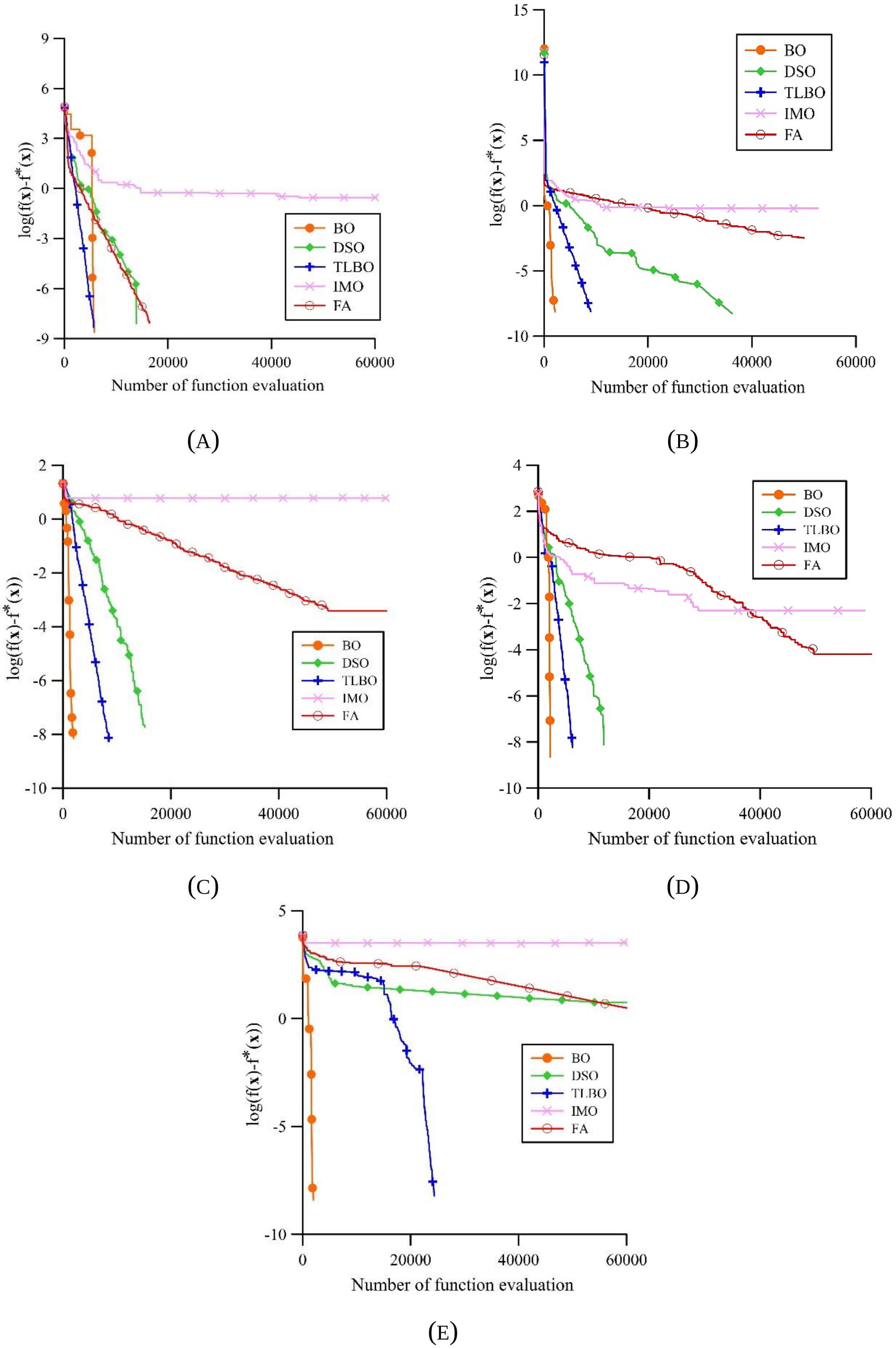


Figure 1. Convergence history for benchmark functions (a) Sphere (b) Schwefel 2.22 (c) Ackley (d) Griewank (e) Rastrigin

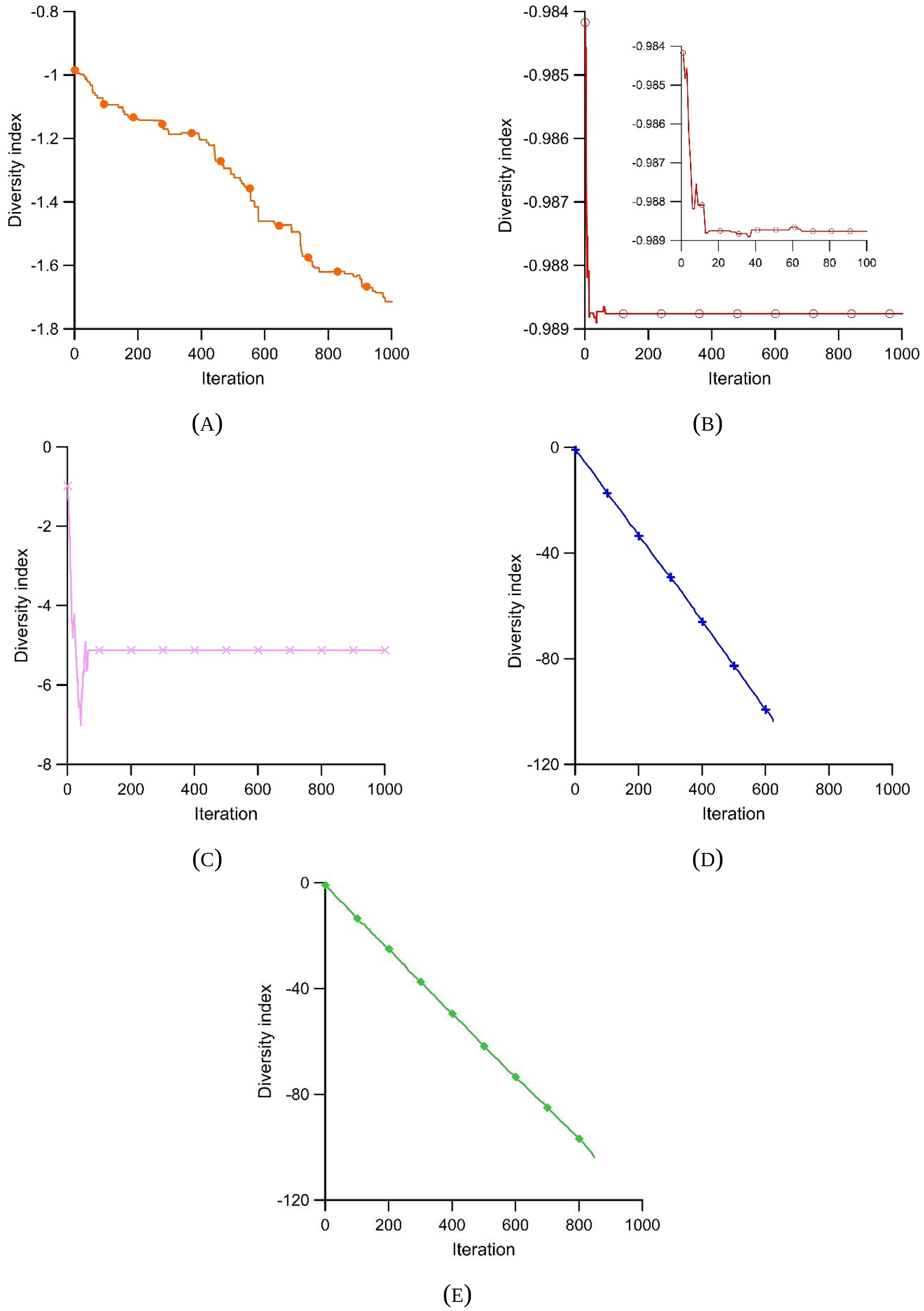


Figure 2. Diversity history for (a) BO, (b) FA, (c) IMO, (d) TLBO and (e) DSO

Complexity analysis

In this section the complexity level of the selected algorithms is comparatively measured. To this end, the formulation suggested in [26] is applied. Based on this formulation to measure the complexity level of an algorithm three different times as T0, T1 and $\hat{T}2$ should be calculated as follows [26]:

- *T0* should be computed via 1000000 times evaluation 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 calculated via 200000 times evaluation of a certain function, which Rastering function is selected in this study for the certain dimensions as *D*=30 and *D*=50.

- *T2* is assessed via time required for complete run of the proposed algorithm over the same function and $\hat{T}2$ is the mean value for five calculated *T2* times.

The complexity level for all five algorithms are calculated on same device and the results are tabulated in Table 9 and Table 10 for Rastering function with *D*=30 and *D*=50, respectively. As can be seen from these tables TLBO method is most complex algorithm while IMO is the least complex method among all algorithms. It should be noted that in these tables the algorithms are ranked for complexity ascendingly.

Table 9. Complexity computation result for selected algorithms for D=30

Algorithm	T0	T1	$\hat{T}2$	($\hat{T}2$-T1)/T0	Rank
FA	1.4E-01	1.9E-01	5.00E+02	3.57E+03	3
TLBO	1.4E-01	1.9E-01	6.00E+02	4.28E+03	4
DSO	1.4E-01	1.9E-01	6.67E+02	4.76E+03	5
IMO	1.4E-01	1.9E-01	3.95E+02	2.82E+03	1
BOA	1.4E-01	1.9E-01	4.60E+02	3.28E+03	2

Table 10. Complexity computation result for selected algorithms for D=50

Algorithm	T0	T1	$\hat{T}2$	($\hat{T}2$-T1)/T0	Rank
FA	1.4E-01	3.05E-01	5.59E+02	3.99E+03	3
TLBO	1.4E-01	3.05E-01	7.14E+02	5.10E+03	4
DSO	1.4E-01	3.05E-01	8.08E+02	5.77E+03	5
IMO	1.4E-01	3.05E-01	4.01E+02	2.86E+03	1
BOA	1.4E-01	3.05E-01	4.73E+02	3.38E+03	2

Constrained benchmark problems

In the current section the performance of the selected methods on handling the constrained optimization is evaluated. However, since the metaheuristic approaches are non-constrained methods, to handling the constraints of the problems in this investigation the penalty function is applied as below:

$$f_{penalty}(\mathbf{X}) = (1+\varepsilon_1 v)^{\varepsilon_2} \times f(\mathbf{X}) \qquad (13)$$
$$v = \sum_{i=1}^{q} \max\{0, g_i(\mathbf{X})\}$$

in which, $f_{penalty}$ is the penalized objective function of the problem and *f*(**X**) is the regular objective function value and $g_i(\mathbf{X})$ returns the i^{th} constraint's violation of the optimization problem. Also, ε_1 and ε_2 are the tuning terms of the penalty function which are taken as 1 and 1.5 while linearly increased up to 6, respectively [9]. For provide a reliable results and preventing any premature convergence the algorithms is permitted to perform 100000 number of OFEs or 500 null iterations (i.e. iteration without improving the solution) whichever occurs first.

Pressure vessel problem

In this section, the optimization of manufacturing cost of the pressure vessel capped in both ends with two parabolic heads shown in Figure 3 is selected as a real world constrained example. The related cost function of the problem and its constraints are formulated in Eqs. (14),(15) . This function combines both costs of the material required for components and necessary welding. The vessel system can work in the 3000 Psi condition while the vessel maximum volume should be limited up to 750 ft^3. The thicknesses of the head parts and shell bodies are restricted up to 1.1 in. and 2 in., respectively. For this problem the decision variables are set as: x1= the shell thickness (T_s), x_2= the head thickness (T_h), x_3= the radius of cylindrical shell (R), and x_4= the length of shell (L). Achieved results for all selected methods are comparatively is tabulated in Table 11. Based on the given information TLBO and DSO can respectively attain the best solutions. After these two methods FA provides a near optimal solution. Interestingly it can be observed that despite of BOA method outperforms all other methods in the non-constrained problems, in the current constrained problem it does not show a significant performance and it stands in fourth place among all five tested methods and IMO ranked in the last place for the current problem. Also, considering statistical data TLBO with lowest value of standard deviation shows most stable behavior.

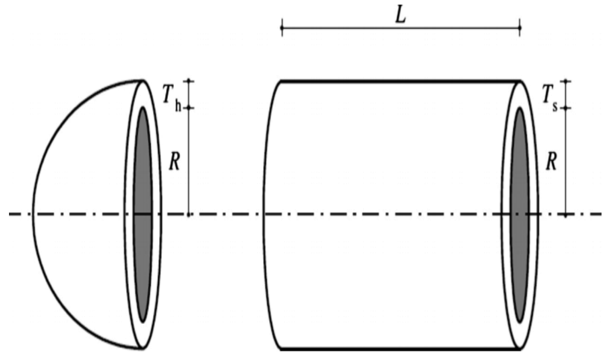


Figure 3. The system of pressure vessel

$$f(\mathbf{X}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \quad (14)$$

while it is constrained as follow:

$$\begin{aligned} g_1(\mathbf{X}) &= -x_1 + 0.0193x_3 \le 0 \\ g_2(\mathbf{X}) &= -x_2 + 0.00954x_3 \le 0 \\ g_3(\mathbf{X}) &= -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \le 0 \\ g_4(\mathbf{X}) &= x_4 - 240 \le 0 \end{aligned} \quad (15)$$

where variables should be bounded as below

$$\begin{aligned} 0 &\le x_1 \le 100 \\ 0 &\le x_2 \le 100 \\ 10 &\le x_3 \le 200 \\ 10 &\le x_4 \le 200 \end{aligned} \quad (16)$$

Table 11. Comparison of the optimal results for the pressure vessel problem

Values	FA	TLBO	DSO	IMO	BOA
$x_1(T_s)$	0.783564	0.778169	0.778167	1.072400	0.812499
$x_2(T_h)$	0.387770	0.384649	0.384649	0.503148	0.437500
$x_3(R)$	40.59555	40.319689	40.31963	52.75704	42.09126
$x_4(L)$	196.30941	199.999022	199.9998	78.56643	176.7465
f(**X**)					
Best	5898.33	5884.71	5884.719	6756.42	6061.077
Worst	6075.45	5889.41	5884.790	6820.33	6363.804
Mean	5998.02	5886.01	5884.728	6788.25	6347.133
Std. Dev.	250.38	3.68	4.91	101.23	326.4545
OFEs	52250	10360	9970	16320	2120

Welded beam problem

Next constrained problem taken into account to measure the search performance of selected optimization methods is the design of the welded beam problem presented in Figure 4. The objective function of the current problem is the total cost minimization of the welded beam satisfying constraints restrict bending stress (σ), deflection (δ) and shear stress (τ). Four aspects associated with the cross-section (b, t) and welds (l, h) are considered as the decision variables. So, this problem mathematically formulated in Eq.(16)-(19). As can be seen the search space of this problem is restricted with seven different constraints. The obtained results for all methods are comparatively tabulated in Table 12. Based on the given information the methods can be sorted as TLBO, FA, DSO, BOA and IMO with 1.724856, 1.73309, 1.737297, 1.90001 and 2.093012 objective values, respectively. Is should be noted that, respecting to the statistical data TLBO with lowest value of standard deviation shows most stable behavior.

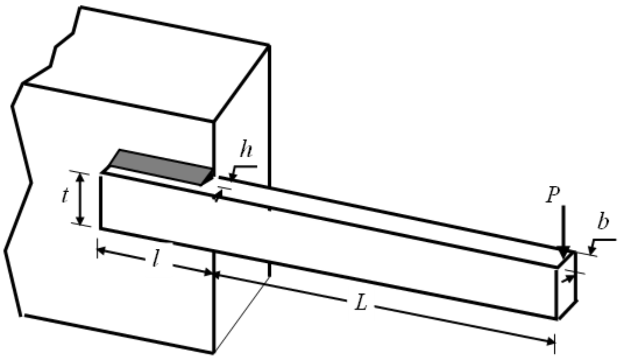


Figure 4. The welded beam

$$\mathrm{cost}(\mathbf{X}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14 + x_2) \quad (17)$$

where,

$\mathbf{X} = \{x_1, x_2, x_3, x_4\}$

subjected to

$$
\begin{aligned}
g_1(x) &= \tau(x) - \tau_{\max} \le 0 \\
g_2(x) &= \sigma(x) - \sigma_{\max} \le 0 \\
g_3(x) &= x_1 - x_4 \le 0 \\
g_1(x) &= 0.10471x_1^2 + 0.04811x_3x_4(14 + x_2) - 5 \le 0 \\
g_5(x) &= 0.125 - x_1 \le 0 \\
g_6(x) &= \delta(x) - \delta_{\max} \le 0 \\
g_7(x) &= p - p_c(x) \le 0
\end{aligned}
\quad (18)
$$

bounded with

$0.1 \le x_1 \le 2$

$0.1 \le x_2 \le 10$

$0.1 \le x_3 \le 10$

$0.1 \le x_4 \le 2$

where

$$
\begin{aligned}
\tau(x) &= \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2} \\
\tau' &= \frac{P}{\sqrt{2}x_1x_2} \\
\tau'' &= \frac{MR}{J} \\
M &= P\left(L + \frac{x_2}{2}\right) \\
R &= \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2} \\
J &= 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\} \\
\sigma(x) &= \frac{6PL}{x_4x_3^2} \\
\delta(x) &= \frac{4PL^3}{Ex_3^3x_4} \\
P_c(x) &= \frac{4.013\sqrt{E(x_3^2x_4^6/36)}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)
\end{aligned}
\quad (19)
$$

where,

$P = 6000$ lb, $L = 14$ in., $E = 30 \times 10^6$ psi,

$\tau_{\max} = 13{,}600$ psi, $\sigma_{\max} = 30{,}000$ psi,

$\delta_{\max} = 0.25$ in.

Table 12. Comparison of the optimal results for welded beam design problem

Values	FA	TLBO	DSO	IMO	BOA
$x_1(h)$	0.206976	0.205730	0.199742	0.206188	0.248729
$x_2(l)$	3.460759	3.470484	3.612060	4.952083	2.953780
$x_3(t)$	8.992902	9.036616	9.037500	8.647560	8.362973
$x_4(b)$	0.207736	0.20573	0.206082	0.235955	0.249008
f(**X**)					
Best	1.73309	1.724856	1.737297	2.093012	1.90001
Worst	1.99654	1.728774	1.994651	7.332687	3.89657
Mean	1.75741	1.730001	1.813290	4.978542	2.55489
Std. Dev.	0.15637	0.070401	0.092100	4.002589	1.99562
OFEs	100000	10680	11680	25860	8650

Design of a spatial 582-bar tower

The last case is devoted to a structural optimization example as the weight minimization of the spatial 582-bar tower shown in Figure 5. To maintain symmetry, the members of the structure are categorized into 32 independent groups. There are three load condition acting on the tower as follows:

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

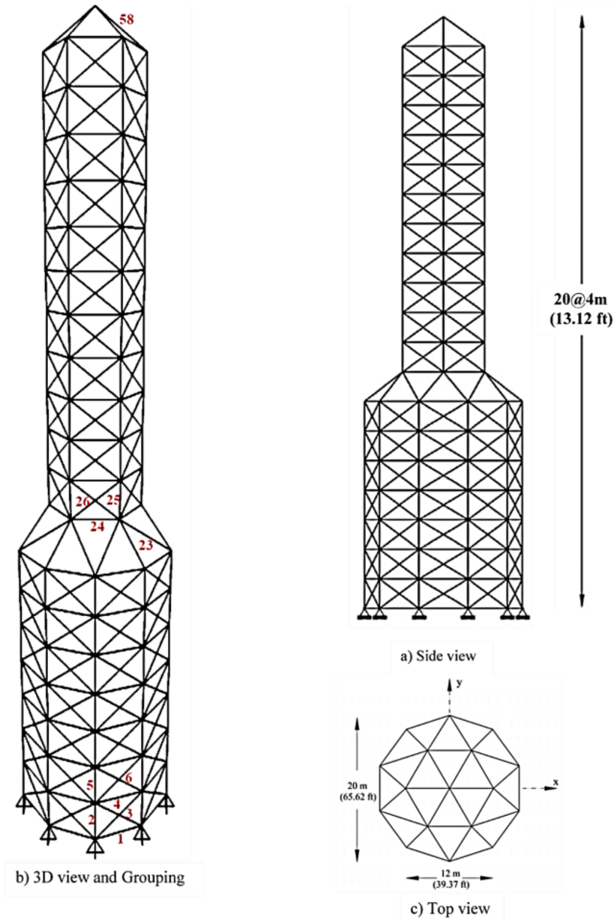


Figure 5. The 582-bar truss tower

The sizing variables for this example are selected from the discrete set addressed in Table 13. The members of this set consist of 140 W-shape profiles given in steel structural profiles of AISC-ASD.

Table 13. 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 and lower bounds are respectively limited to 6.16 in^2 (39.74 cm^2) and 215.00 in^2 (1387.09 cm^2). The nodal displacement for all principal directions are limited up to 3.15 in (8 cm). The stress limitation is determined based on the buckling criterion of the AISD-ASD89 code as follows [1]:

$$\begin{cases} \sigma_i^+ = 0.6F_y & \sigma_i \geq 0 \\ \sigma_i^- & \sigma_i < 0 \end{cases} \quad (20)$$

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

$$\sigma_i^- = \begin{cases} \left[\left(1-\frac{\lambda_i^2}{2C_c^2}\right)F_y \Big/ \left(\frac{5}{3}+\frac{3\lambda_i}{8C_c}-\frac{\lambda_i^3}{8C_c^3}\right)\right] & for\ \lambda_i < C_c \\ \frac{12\pi^2 E}{23\lambda_i^2} & for\ \lambda_i \geq C_c \end{cases} \quad (21)$$

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

$$C_c = \sqrt{\frac{2\pi^2 E}{F_y}} \quad (22)$$

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

$$\lambda_i = \frac{k_i l_i}{r_i} \leq \begin{cases} 300\ for\ tension\ members \\ 200\ for\ compression\ members \end{cases} \quad (23)$$

where λ_i, r_i, and li are the slenderness ratio, radius of gyration and length of the i^{th} member, respectively. For compression elements, if the required slenderness ratio is not be satisfied, the allowable stress must not surpass the value of $\left(\frac{12\pi^2 E}{23\lambda_i^2}\right)$ ever [1]. This example as the complex structural optimization are solved via five selected techniques and the archived results are tabulated in Table 14. The number of structural analyses (NSA) and standard deviation (STD) for each algorithm are provided.

Table 14. Comparison of the optimal results for the 582-bar tower problem

Groups	Optimal cross-sectional areas				
	FA	TLBO	DSO	IMO	BO
1	W8×21	W8×21	W8×24	W8×21	W8×24
2	W24×76	W24×84	W12×72	W12×79	W24×68
3	W8×21	W8×21	W8×28	W8×24	W8×28
4	W12×65	W24×62	W12×58	W10×60	W18×60
5	W8×21	W8×21	W8×24	W8×24	W8×24
6	W8×21	W8×21	W8×24	W8×21	W8×24
7	W10×54	W16×57	W10×49	W8×48	W21×48
8	W8×21	W8×21	W8×24	W8×24	W8×24
9	W8×21	W8×21	W8×24	W8×21	W10×26
10	W12×50	W12×53	W12×40	W10×45	W14×38
11	W8×21	W8×21	W12×30	W8×24	W12×30
12	W10×68	W10×77	W12×72	W10×68	W12×72
13	W24×76	W21×83	W18×76	W14×74	W21×73
14	W14×53	W21×57	W10×49	W8×48	W14×53
15	W12×79	W18×76	W14×82	W18×76	W18×86
16	W8×21	W8×21	W8×31	W8×31	W8×31
17	W12×65	W10×22	W14×61	W8×21	W18×60
18	W8×21	W18×55	W8×24	W16×67	W8×24
19	W8×21	W8×21	W8×21	W8×24	W16×36
20	W12×45	W8×21	W12×40	W8×21	W10×39
21	W8×21	W14×30	W8×24	W8×40	W8×24
22	W8×21	W8×21	W14×22	W8×24	W8×24
23	W16×26	W8×21	W8×31	W8×21	W8×31
24	W8×21	W8×21	W8×28	W10×22	W8×28
25	W8×21	W8×21	W8×21	W8×24	W8×21
26	W8×21	W8×21	W8×21	W8×21	W8×24
27	W8×21	W10×22	W8×24	W8×24	W8×28
28	W8×21	W8×21	W8×28	W8×24	W14×22
29	W8×21	W8×21	W16×36	W8×24	W8×24
30	W8×21	W8×31	W8×24	W8×24	W8×24
31	W8×21	W8×21	W8×21	W8×24	W14×22
32	W8×21	W12×22	W8×24	W8×24	W8×24
Volume (m^3)	20.07	20.3	22.07	23.4	22.37
NSA	25890	16050	17670	5850	8880
Std. (m^3)	0.53	0.22	0.51	1.82	0.32

Conclusion

The current investigation deals with comparatively measurement of the five metaheuristic algorithms from various aspects. These algorithms have been selected to cover the latest methods announced in the last decade (between 2009 to 2019). The selected algorithms chronologically (i.e. based on their emerging date) can be sorted as Firefly Algorithm (FA), Teaching and Learning Based Optimization (TLBO), Drosophila Food-Search Optimization (DSO), Ions Motion Optimization (IMO) and Butterfly Optimization Algorithm (BOA). To gain a comprehensive assessment, the performance of these algorithms are verified on different classes of optimization problems contain mathematical, mechanical and structural cases. Beside of the variety, technically these cases consist both constraint and non-constrained search spaces, discrete and continuous design variables. So, they comprehensively challenge the algorithms on handling the different types of domains and boundaries. The methods are evaluated and compared considering different features like convergence rate, stability, diversity, complexity and accuracy level.

The achieved outcomes are provided through the illustrative tables and diagrams. Obtained numeric result show that BOA could outperform all other algorithms on handling the non-constrained optimization problems. But interestingly it cannot show the same performance on handling the constrained optimization problems, and in this category TLBO demonstrates the superior performance. Complexity tests reveal that DSO stands as the most complex method while IMO is the least complex algorithm among the all other selected techniques. Also, it should be noted that for the constrained problems, TLBO with the least standard deviation value shows the most stability level on finding the optimal solution. Overall insight into the results declares that not necessarily the latest method(s) should be picked up as the best method(s), but the algorithms, as the black-box optimizer tools, may show distinctive performances on different classes of problems. Consequently, since this investigation cover different classes of optimization problems, it provides contributory platform for scholars on taking the most proper algorithm(s) for desired problems.

References

1. AISC American Institute of Steel Construction (AISC). Manual of steel construction allowable stress design. 9th ed. Chicago, IL. **1989**, 1-360,
2. Arora, S.,Singh, S. Butterfly optimization algorithm: a novel approach for global optimization. *Soft Computing* **2019**, 23(3), 715-734, http://dx.doi.org/10.1007/s00500-018-3102-4
3. Camp, C.V.,Farshchin, M. Design of space trusses using modified teaching–learning based optimization. *Engineering Structures* **2014**, 62–63(87-97, http://dx.doi.org/10.1016/j.engstruct.2014.01.020
4. Das, K.N.,Singh, T.K. Drosophila Food-Search Optimization. *Applied Mathematics and Computation* **2014**, 231(566-580, http://dx.doi.org/10.1016/j.amc.2014.01.040
5. Degertekin, S.O.,Hayalioglu, M.S. Sizing truss structures using teaching-learning-based optimization. *Computers & Structures* **2013**, 119(177-188, https://doi.org/10.1016/j.compstruc.2012.12.011
6. Eker, M.K.,Karadeniz, A. Rüzgar Ve Termik Santrallerden Oluşan Enerji Sistemlerinde Ekonomik Güç Dağılımının Big-Bang Big-Crunch, PSO ve IMO Algoritmaları İile İrdelenmesi. *Politeknik Dergisi* **2016**, 19(3),
7. 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* **2010**, 88(17–18), 1033-1048, http://dx.doi.org/10.1016/j.compstruc.2010.06.006
8. Javidy, B.,Hatamlou, A.,Mirjalili, S. Ions motion algorithm for solving optimization problems. *Applied Soft Computing* **2015**, 32(72-79, http://dx.doi.org/10.1016/j.asoc.2015.03.035
9. Kaveh, A.,Zolghadr, A. Democratic PSO for truss layout and size optimization with frequency constraints. *Computers & Structures* **2014**, 130(10-21, https://doi.org/10.1016/j.compstruc.2013.09.002
10. Kennedy, J.,Eberhart, R., 1995. Particle swarm optimization, Neural Networks, 1995. Proceedings., IEEE International Conference on. pp. 1942-1948 vol.1944.
11. Lieu, Q.X.,Do, D.T.T.,Lee, J. An adaptive hybrid evolutionary firefly algorithm for shape and size optimization of truss structures with frequency

constraints. *Computers & Structures* **2018**, 195(99-112, https://doi.org/10.1016/j.compstruc.2017.06.016

12. Lim, W.H.,Mat Isa, N.A. Two-layer particle swarm optimization with intelligent division of labor. *Engineering Applications of Artificial Intelligence* **2013**, 26(10), 2327-2348, http://dx.doi.org/10.1016/j.engappai.2013.06.014
13. Lim, W.H.,Mat Isa, N.A. Teaching and peer-learning particle swarm optimization. *Applied Soft Computing* **2014**, 18(39-58, http://dx.doi.org/10.1016/j.asoc.2014.01.009
14. Miguel, L.F.F.,Lopez, R.H.,Miguel, L.F.F. Multimodal size, shape, and topology optimisation of truss structures using the Firefly algorithm. *Advances in Engineering Software* **2013**, 56(23-37, http://dx.doi.org/10.1016/j.advengsoft.2012.11.006
15. Moloodpoor, M.,Mortazavi, A.,Ozbalta, N. Thermal analysis of parabolic trough collectors via a swarm intelligence optimizer. *Solar Energy* **2019**, 181(264-275, https://doi.org/10.1016/j.solener.2019.02.008
16. Mortazavi, A. Interactive fuzzy search algorithm: A new self-adaptive hybrid optimization algorithm. *Engineering Applications of Artificial Intelligence* **2019**, 81(270-282, https://doi.org/10.1016/j.engappai.2019.03.005
17. Mortazavi, A.,Toğan, V. Triangular units based method for simultaneous optimizations of planar trusses. *Advances in Computational Design* **2017**, 2(3), 195-210, https://doi.org/10.12989/acd.2017.2.3.195
18. Mortazavi, A.,Togan, V.,Daloğlu, A.,Nuhoglu, A., 2016. Simultaneous topology and sizing optimization of trusses with two different optimization algorithms, 12th International Congress on Advances in Civil Engineering. Istanbul, Turkey.
19. Mortazavi, A.,Toğan, V.,Moloodpoor, M. Solution of structural and mathematical optimization problems using a new hybrid swarm intelligence optimization algorithm. *Advances in Engineering Software* **2019**, 127(106-123, https://doi.org/10.1016/j.advengsoft.2018.11.004
20. Mortazavi, A.,Toğan, V.,Nuhoğlu, A. An integrated particle swarm optimizer for optimization of truss structures with discrete variables. *Structural Engineering and Mechanics* **2017**, 61(359-370, https://doi.org/10.12989/sem.2017.61.3.359
21. 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* **2017**, 23(8), 985-1001, http://dx.doi.org/10.3846/13923730.2017.1348982
22. 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* **2011**, 43(3), 303-315, http://dx.doi.org/10.1016/j.cad.2010.12.015
23. Rao, R.V.,Waghmare, G.G. Complex constrained design optimisation using an elitist teaching-learning-based optimisation algorithm. *International Journal of Metaheuristics* **2014**, 3(1), 81-102, http://dx.doi.org/10.1504/ijmheur.2014.058863
24. Ray, T.,Liew, K.M. Society and civilization: An optimization algorithm based on the simulation of social behavior. *IEEE Transactions on Evolutionary Computation* **2003**, 7(4), 386-396, http://dx.doi.org/10.1109/TEVC.2003.814902
25. Storn, R.,Price, K. Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization* **1997**, 11(4), 341-359, http://dx.doi.org/10.1023/a:1008202821328
26. 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,
27. Toğan, V. Design of planar steel frames using Teaching–Learning Based Optimization. Engineering Structures 2012, 34(225-232, http://dx.doi.org/10.1016/j.engstruct.2011.08.035
28. Toğan, V.,Mortazavi, A. Sizing optimization of skeletal structures using teaching-learning based optimization. Optimization and Control: Theories Applications 2017, 7(2), 12, http://dx.doi.org/10.11121/ijocta.01.2017.00309
29. Yang, X.-S., 2009. Firefly Algorithms for Multimodal Optimization, in: Watanabe, O., Zeugmann, T. (Eds.), Stochastic Algorithms: Foundations and Applications. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 169-178.
30. Yuhui, S.,Eberhart, R.C., 2001. Fuzzy adaptive particle swarm optimization, Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546). pp. 101-106 vol. 101.