Journal of Structural Engineering & Applied Mechanics (2023) 6(4):385-406
DOI 10.31462/jseam.2023.04385406

RESEARCH ARTICLE

Swarm-based search algorithms: A comparative study on sizing optimization of truss structures

Ali Mortazavi[1*]

[1] Izmir Democracy University, Department of Civil Engineering, Izmir, Türkiye

Article History

Received 3 June 2023
Accepted 4 August 2023

Keywords

Optimization techniques
Metaheuristic algorithms
Structural sizing optimization
Truss systems

Abstract

Metaheuristic algorithms belong to the category of non-deterministic optimization methods. Extensive research conducted in this domain has elucidated that each of these methods possesses distinctive merits and demerits. To illustrate, one algorithm may exhibit a notable penchant for exploration, whereas another algorithm may demonstrate exceptional prowess in exploitation. The judicious selection of a suitable and efficient algorithm for a given problem can profoundly influence both the rate of convergence and the level of accuracy. Over the past few decades, various swarm-based metaheuristic algorithms have been introduced in technical literature. Consequently, undertaking a comprehensive comparative evaluation of a subset of these methodologies can furnish researchers with an essential framework to discern the most appropriate algorithm for their objectives. The current investigation focuses on evaluating and comparing four swarm-based metaheuristic algorithms especially on solving size optimization of truss structures for this purpose. To cover the research conducted in the past two decades as comprehensively as possible, a hierarchical selection process was employed for choosing the methods. As a result, the following algorithms were chosen Firefly Algorithm (FA), Drosophila Food-search Algorithm (DFO), Harris Hawk Optimization (HHO), and Butterfly Optimization Algorithm (BOA). Various characteristics of the selected algorithms, such as convergence rate, diversity variation, complexity, and accuracy of the final solutions, were compared. The findings indicate Harris Hawk Optimization (HHO) could nearly outperform the other selected methods on solving structural size optimization problems.

1. Introduction

In numerous science and engineering domains, the primary objective of the design process is to create a system that maximizes efficiency and minimizes cost. As a result, design problems can be transformed into optimization problems. In this regard, selecting a proper optimization technique playing a crucial role in their solution process [1, 2]. Optimization techniques can be broadly classified into two categories: deterministic and non-deterministic algorithms. Deterministic approaches refer to methods that rely on mathematical modeling and programming techniques. These approaches start the search process from an initial design point then by computing the gradient of the objective function and exploring the search space tries to move towards the optimal point of the search domain. Despite having a fast convergence rate and high accuracy,

* Corresponding author (ali.mortazavi@idu.edu.tr)
eISSN 2630-5763

these approaches have limitations. On one hand, they are highly dependent on the starting point. On the other hand, they require a continuous (or partially continuous) objective function and its gradient information, which can be difficult or impossible to find for many complex engineering problems [3, 4].

To overcome the aforementioned limitations, a different class of methods might be required. Non-deterministic approaches have emerged as an alternative optimization strategy in this regard. These approaches do not rely on gradient information of the objective function or its constraints, if any [5]. Instead, they typically use probabilistic rules of transition. A number of randomly generated agents are employed, and they are gradually improved until the convergence/termination condition is met. Metaheuristic approaches, which are the main techniques belonging to this group, are generally inspired by natural phenomena. These methods model natural concepts such as the survival behavior of animal colonies and physical rules [6]. Since these methods are numeric-based, the usage of these algorithms on different classes of optimization problems has piqued the interest of researchers [7-10]. Metaheuristic methods based on their inspiration origins can be divided into four main classes as given below:

Evolution-inspired algorithms: Evolution-inspired algorithms mimic the patterns of evolution observed in living organisms. Two prominent examples of evolution-inspired algorithms are Genetic Algorithm (GA), which mimics the principles of Darwinian evolution theory [11], and Evolutionary Strategies (ESs), which are based on the general concept of the evolutionary phenomenon [12].

Physics-inspired algorithms: Physics-inspired algorithms simulate existing physical principles. Some examples of these methods include Ion Motion Algorithm (IMO), which mimics the interaction between anions and cations [13]; Heat Transfer Search (HTS) method, which is inspired by heat transfer mechanisms in different environments [14].

Human-inspired algorithms: The techniques belonging to this group simulate various human social or emotional behaviors. For instance, Tabu Search (TS) algorithm restricts the feasible neighborhood candidates by incorporating them in a list [15]. Teaching and Learning Based Optimization (TLBO) imitates the learning process in a classroom [16].

Swarm-inspired algorithms: This group comprises of methods that mimic the survival behaviors of animals in nature. Particle Swarm Optimizer (PSO) mimics the collective behavior of flocks birds in food-searching or escaping from enemies [17], Ant Colony Optimizer (ACO), models the ants food-search strategy [18], Firefly Algorithm (FA), imitates the mating mechanism of the firefly insect [19], Symbiotic Organisms Search (SOS), simulates the organisms symbiotic surviving interaction pattern [20], Drosophila Food-search Optimization (DFO), This method imitates the way Drosophila insects search for food [21], Virus Optimization Algorithm (VOA), Simulates the behavior of virus microorganisms in their struggle for survival [22] and Butterfly Optimization Algorithm (BOA) simulates the food search and mating strategy of butterflies [23] are some examples for this group of optimization methods.

Based on the no-free-lunch theorem, a method (or class of methods) can effectively solve a specific problem but may not perform equally well on other case(s). So, to provide a more realistic perspective on their search capability, they should comparatively be assessed on desired problem. As instance, Mortazavi and Togan conducted tests on the different swarm-inspired, physics-inspired and human-inspired algorithms on size and layout optimization of truss systems [24], they also verified the performance of the same group of metaheuristics (plus a hybrid method) on distinct mechanical, structural and construction management problems [25, 26]. Mortazavi et al. make a comparison between swarm-inspired human-inspired on simultaneous size, shape and topology optimization of truss systems [27]. In the current study, to provide a more specific and accurate assessment of the search capability of the metaheuristic methods on sizing optimization of the truss structures just the swarm-based methods are examined in solving this class of problems.

As stated, the number of swarm-based methods is significantly higher than the other three categories. Therefore, in the current study, four distinct swarm-based metaheuristic algorithms have been selected to evaluate their search performance in solving size optimization of truss structures. The selected algorithms hierarchically can be sorted as, Firefly Algorithm (FA) [19], Drosophila Food-search Optimization (DFO) [21], Harris Hawk Optimization (HHO) algorithm [28] and Butterfly Optimization Algorithm (BOA) [23]. To offer a comprehensive evaluation, the current study incorporates both constrained and unconstrained optimization problems. Furthermore, the performance of the methods is assessed based on various aspects, including convergence rates, accuracy of optimal solutions, statistical outcomes, and behavioral diagrams.

The rest of this study is arranged as follows: Section 2 gives details about the selected optimization techniques. Section 3 provides the performance of the algorithms on handling the unconstrained and constrained structural optimization problems. Section 4 is devoted to providing a brief conclusion.

2. Optimization methods

In this section, selected four different optimization algorithms are described in detail. All these algorithms are population-based and can be started from arbitrary locations in problem's search space. These algorithms employ their own search patterns to guide the population towards the most promising regions of the search domain in order to obtain optimal or near-optimal solutions for the problems at hand.

2.1. Firefly algorithm (FA)

The firefly algorithm (FA) is originally introduced by Yang [19]. The search patterns of this method are derived from the behavior of fireflies in the natural environment. The method is based on the flashing behavior of fireflies and its impact on their own species. To develop a mathematical model, following idealizations are made to simplify the complex behavior of fireflies in reality:

a) The first idealization in this method is that all fireflies are assumed to be a unisex species, which means that regardless of their gender, they can be attracted to each other.
b) 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.
c) The brightness of fireflies is highly affected by the landscape of the objective function of the optimization problem. For instance, in foggy weather, even fireflies in proximity may not be able to perceive each other.

The two crucial elements of the firefly algorithm are the light intensity of the agents and an appropriate formulation to determine the attraction level of the agents. The attraction of other fireflies towards the light intensity of a given firefly is dependent on their distance, which can be expressed through an exponential function as follows:

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

where, r_{ij} parameter represents the distance between two fireflies and can be defined using the Euclidean distance formula which is formulated as $r_{ij} = \|\boldsymbol{X_i} - \boldsymbol{X_j}\|$. Also, γ and β_0 indicate the light absorption and attractiveness coefficients, respectively. Consequently, based on the given information the movement of the fireflies is mathematically formulated as follow:

$$\Delta \boldsymbol{x}_i = \beta_0 e^{-\gamma r_{ij}^2}(x_i - x_j) + \alpha(rand - 0.5)$$
$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \Delta \boldsymbol{x}_i \tag{2}$$

where, the updated location of the firefly is denoted by superscript $t+1$ and t, respectively. The parameter α is a random number generated uniformly from the interval [0,1]. The function 'rand' produces a vector of random numbers selected from the interval [0,1], and the parameter β_0 is set to be 1 [19]. Finally, to enhance clarity, the pseudo code for Firefly Algorithm (FA) is presented in Table 1.

2.2. Drosophila food-search algorithm (DFO)

Das and Singh in 2014 introduced a population-based algorithm called Drosophila Food-search Optimization (DFO), which emulates the food search behavior of the Drosophila Melanogaster insect. This optimization method models the efficient food search strategies of fruit flies and can be applied to optimize complex systems in various fields including engineering, computer science, and mathematics [21]. The mathematical model for DFO method is given below:

$$U_{i,k} = V_{i,k} + \left|V_{r3,k} - V_{r4,k}\right|$$
$$W_{i,k} = V_{i,k} + \left|V_{r3,k} - V_{r4,k}\right| \text{ for } k = r_1 \text{ and } r_2$$
$$\text{for } j \neq r_1 \text{ and } j \neq r_2,\ U_{i,k} = V_{i,j} \text{ and } W_{i,j} = V_{i,j} \tag{3}$$
$$V'_{i,j} = Min\{f(V_{i,j}), f(U_{i,j}), f(W_{i,j})\} \text{ for } \begin{cases} i = 1, 2, \ldots, P \\ j = 1, 2, \ldots, D \end{cases}$$

where, P and D stand for population size and dimension of the problem, respectively, are random scalars selected from $[1, D]$ interval. Also, $V_{i,k}$ and $V'_{i,j}$ are the current and updated agent's location. If in each iteration the agent is not updated (the better agent in not found) the algorithm makes a neighborhood search using the Quadratic Approximation search as below:

$$Child = \frac{1}{2}\frac{(R_2^2 - R_3^2)f(R_1) + (R_3^2 - R_1^2)f(R_2) + (R_1^2 - R_2^2)f(R_3)}{(R_2 - R_3)f(R_1) + (R_3 - R_1)f(R_2) + (R_1 - R_2)f(R_3)} \tag{4}$$

Table 1. The pseudo code for FA

```
Initialize the internal parameters of the algorithm
Generate a population with n random fireflies
Determine Light intensity I_i at X_i which by computing f(X_i)
while (termination conditions are not met)
  for (each firefly i)
    for (each firefly j)
      if (I_i > I_j, ith firefly should move towards jth firefly
        Determine attractiveness coefficient based on the distance between two fireflies
        Compute the objective function (light intensity)
      end
    end
  end
end
```

Table 2. The pseudo code for DFO

Initialize internal parameters
Generate n random agents
Evaluate objective function value of each agent
while (termination conditions are not met)
for (each agent i)
update the agent's locations using Eq. (3)
if the new agent is improved take if not reject it
if the new and old objective value of the agent is within 1%, make a local search using Eq. (4)
if the new agent is improved take if not reject it
end
end

in which $f(.)$ denotes the value of problem's objective function. R_1, R_2 and R_3 are three agents randomly selected among the colony ($R_1 \neq R_2 \neq R_3$). For more precision, the pseudo code for this method is provided in Table 2.

2.3. Harris hawks optimizer (HHO)

The Harris Hawks Optimizer (HHO) is a non-gradient and swarm-based optimization approach that draws inspiration from the hunting behavior and chasing techniques of Harris Hawks. The algorithm emulates the bird's surprise pounces to catch prey. Based on the Hawk's behavior in nature the HHO operates three distinct search phases: Exploration phase, transition phase, and exploitation phase. The phases are mathematically formulated as follows [28]:

Exploration phase

$$X^{t+1} = \begin{cases} X_j^t - r_1\left|X_j^t - 2r_2X^t\right| & q \geq 0.5 \\ \left(X_{prey}^t - X_m^t\right) - r_3\left(LB + r_4(UB - LB)\right) & q < 0.5 \end{cases}$$
$$X_m^t = \frac{1}{N}\sum_{i=1}^{N} X_i^t \tag{5}$$

Transition from exploration to exploitation:

$$E = 2E_0\left(1 - \frac{t}{T}\right) \tag{6}$$

Exploitation phase:

for $r \geq 0.5$ and $|E| \geq 0.5$

$$X^{t+1} = \Delta X^t - E\left|JX_{prey}^t - X^t\right|$$
$$\Delta X^t = X_{prey}^t - X^t \tag{7}$$
$$J = 2(1 - r_5)$$

for $r \geq 0.5$ and $|E| < 0.5$

$$X^{t+1} = X_{prey}^t - E|\Delta X^t|$$

for $r < 0.5$ and $|E| \geq 0.5$

$$Y = X_{prey}^{t} - E\left|JX_{prey}^{t} - X^{t}\right|$$

$$Z = Y + S \times LF(D)$$

$$LF(x) = 0.01 \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}} \quad , \quad \sigma = \left(\frac{\tau(1+\beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\tau\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}}\right)^{\frac{1}{\beta}}$$

$$X^{t+1} = \begin{cases} Y & if \quad f(Y) < F(X^{t}) \\ Z & if \quad f(Z) < F(X^{t}) \end{cases}$$

for $r < 0.5$ and $|E| < 0.5$

$$X^{t+1} = \begin{cases} Y & if \quad f(Y) < F(X^{t}) \\ Z & if \quad f(Z) < F(X^{t}) \end{cases}$$

$$Y = X_{prey}^{t} - E\left|JX_{prey}^{t} - X_{m}^{t}\right|$$

where, r_n (n = 1, 2, 3, 4) and q are scalars randomly selected from [0, 1] interval. Also, X^{t+1} and X^{t} are the updated and current locations of the agents, respectively; X_{prey}^{t} is a vector depicts the location of the prey; X^{t} shows the current location of the predator; the upper and lower bounds of the selected variables are respectively shown by UB and LB; X_{j}^{t} indicates a randomly selected agent; the arithmetic mean of the all agents is shown by X_{m}^{t}; the population size is shown by N. The pseudo code for this method is given in Table 3.

2.4. Butterfly optimization algoritm (BOA)

The Butterfly Optimization Algorithm (BOA) imitates the mating and food-finding behavior of butterflies to search the problem's domain. In this method, butterflies emit fragrances to attract one another, and subsequently, each butterfly randomly moves towards the best butterfly in the colony. The characteristics of the search space determine the level of stimulus intensity for each butterfly. The BOA encompasses distinct strategies for global and local searches, and its mathematical formulation is as follows [23]:
Global search:

$$x_{i}^{t+1} = x_{i}^{t} + (r^{2} \times g^{*} - x_{i}^{t}) \times f_{i} \tag{8}$$

Local search:

$$x_{i}^{t+1} = x_{i}^{t} + \left(r^{2} \times x_{j}^{t} - x_{k}^{t}\right) \times f_{i} \tag{9}$$

where

$$f_{i} = Cf^{a}$$

where, g^* denotes the colony's best agent; x_j and x_k are two agents selected randomly from entire colony; superscripts of t and $t+1$ indicate the current and updated states of the parameter, respectively. Also, r is a random scalar picked from interval of [0,1]; The fragrance factor (f) specifies the objective function value of the desired agent, while C and a are two tuning parameters which are taken as 0.001 and 0.1 [23]. For more clarity the pseudo code for the BOA method is given in Table 4.

3. Numerical examples

In this section, the search performance of the selected methods is firstly tested on some unconstrained mathematical optimization problems to provide general perspective on the selected algorithm and their search capability are comparatively evaluated on solving sizing optimization of truss systems. The computations were carried out on a system featuring an intel CORE i7@2GHz processor and 16GB of RAM. To ensure that premature convergence did not occur, each case was executed 30 times.

Table 3. The pseudo code for HHO method

```
Setup agents and internal parameters
while (termination conditions are not met)
  compute the objective function value of agents
  assign X^t_prey as the best location
    for (each agent)
        update E_0 and J and E utilizing Eqs. (5-7)
        employ r and E values for determining algorithm's phase using Eqs. (5-7)
        update the location vector utilizing Eqs. (5-7)
    end
end
```

Table 4. The pseudo code for BOA

```
Initialize parameters
Generate n random agents (butterflies)
while (termination conditions are not met)
    for each agent (butterfly)
          calculate the objective function for current butterfly
    end
   Select the best agent
   for each agent
         Generate random scalar (r)
         If r < p then
              Make a global search using Eq. (8)
         else
             Make a global search using Eq. (9)
         end
    end
end
```

3.1. Unconstrained benchmark functions

The purpose of this section is to evaluate the efficacy of the chosen algorithms in optimizing unconstrained benchmark functions. To accomplish this, ten of the most prominent benchmark functions with varying properties have been chosen. The list of these functions and their formulations are given in Table 5. For more clarity, the properties of these functions are also given in Table 6. The first five functions have the optimum value in the origin while the other functions have bias from the origin.

3.1.1. Convergence rate analysis

In this section, the convergence rate of the selected algorithms is verified on six unconstrained mathematical problems. These problems and their specifications and boundaries are given in Tables 5 and 6, respectively [29]. The convergence diagrams for these functions are provided in Fig. 1. Based on these diagrams, the HHO method overly shows a better performance in comparison with other selected techniques. It should be noted that, in the case of Schwefel's Problem 1.2 with Noise in Fitness function the FA method can outperform other techniques. This indicates that the performance of metaheuristic methods, regardless of whether they were introduced recently or in the past, can vary across different problems. Considering these results, conducting such comparisons can provide valuable information for selecting the appropriate method before solving a problem.

Table 5. Benchmark functions

No	Function	Formulation
F1	Sphere	$f(\mathbf{X}) = \sum_{i=1}^{D} x_i^2$
F2	Rastrigin	$f(\mathbf{X}) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$
F3	Schwefel 2.21	$f(\mathbf{X}) = \max_{i=1,\dots,D} \lvert x_i \rvert$
F4	Rosenbrock	$f(\mathbf{X}) = \sum_{i=1}^{D-1} \left(100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right)$
F5	Shifted Rotated Elliptic	$f(\mathbf{X}) = \sum_{i=1}^{D} (10^6)^{\frac{i-1}{D-1}} z_i^2 - 450$
F6	Shifted Schwefel's Problem 1.2 with Noise in Fitness	$f(\mathbf{X}) = \left(\sum_{i=1}^{D}\left(\sum_{j=1}^{i} z_j\right)^2\right) * (1 + 0.4\lvert N(0,1)\rvert) - 450$

Table 6. Properties of the selected functions

No	Features	Range	f_{min}	D
F1	US	[-100, 100]	0	30
F2	UN	[-10, 10]	0	30
F3	UN	[-100, 100]	0	30
F4	UN	[-100, 100]	0	30
F5	UHNRC	[-100, 100]	-450	30
F6	UHNOC	[-100, 100]	-450	30

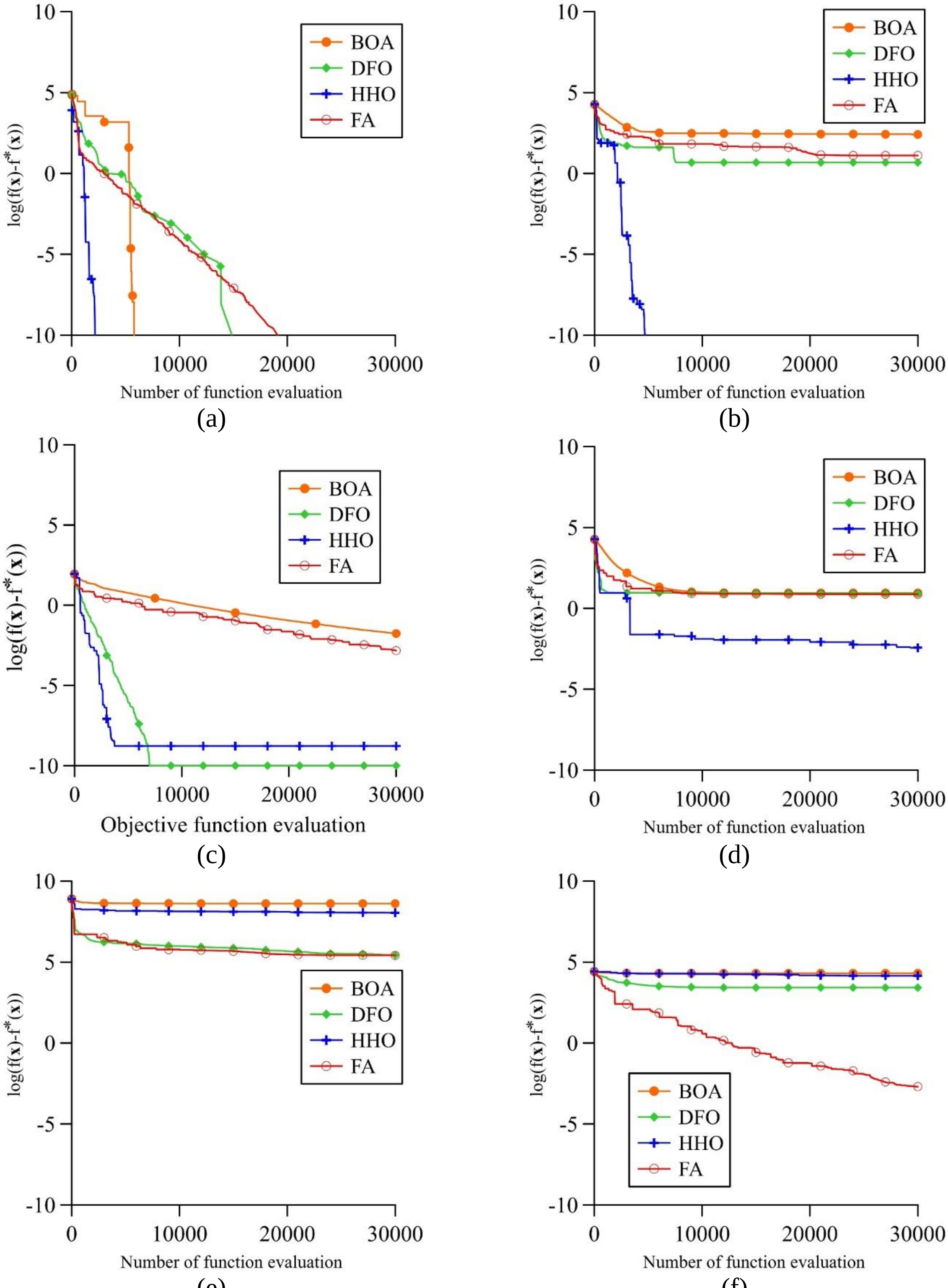


Fig. 1. The convergence diagrams for (a) Sphere, (b) Rastrigin, (c) Schwefel 2.21, (d) Rosenbrock, (e) Shifted Rotated Elliptic and (f) Shifted Schwefel's Problem 1.2 with Noise in Fitness

3.1.2. Complexity analysis

In this section, we assess the relative complexity of the chosen algorithms by employing the formulation proposed in [29]. According to this formulation, three distinct time metrics, namely T_0, T_1, and $\hat{T}_2$, are calculated to quantify the complexity level of an algorithm. These time metrics are calculated as follows:

To determine T_0, the computation involves running a loop 1,000,000 times, evaluating the runtime for each iteration. The loop is as follows:

$$\boldsymbol{for}\ i = 1{:}\,1000000$$
$$x = 5.55\ (x\ is\ double);$$
$$x = x + x;\ x = x/2;\ x = x * x;\ x = sqrt(x);\ x = ln(x);\ x = exp(x);\ y = x/x;$$
$$\boldsymbol{end}$$

For T_1, the calculation involves evaluating a specific function, namely the Rastrigin function, 200,000 times for different dimensions, such as $D = 30$ and $D = 50$. T_2 is determined by measuring the time it takes for the proposed algorithm to run completely on the same function. $\hat{T}_2$ is the average value obtained from five separate measurements of T_2.

The complexity levels of all four algorithms were computed on the same device, and the results are presented in Table 8 and Table 9 for the Rastrigin function with dimensions $D = 30$ and $D = 50$, respectively. From these tables, it is evident that the DFO method exhibits the highest complexity among the algorithms, while the BOA method is the least complex. It is important to note that the algorithms in these tables are ranked in ascending order based on complexity.

3.1.3. Behavioral analysis

In this section, we evaluate the performance of the selected optimization methods by analyzing its search behavior through the generation of behavioral diagrams in Fig. 2. This approach utilizes the concept of diversity within the population to determine whether the algorithm is in an exploration or exploitation mode. To achieve this, it is important to provide a definition for the diversity concept. Mathematically, the diversity level within the population can be defined as follows [30]:

Table 8. Complexity computation result for selected algorithms for $D = 30$

Algorithm	T_0	T_1	$\hat{T}_2$	$(\hat{T}_2 - T_1)/T_0$	Rank
FA	1.4E-01	1.9E-01	5.00E+02	3.57E+03	2
DFO	1.4E-01	1.9E-01	6.67E+02	4.76E+03	4
HHO	1.4E-01	1.9E-01	6.10E+02	4.355E+03	3
BOA	1.4E-01	1.9E-01	4.60E+02	3.28E+03	1

Table 9. Complexity computation result for selected algorithms for $D = 50$

Algorithm	T_0	T_1	$\hat{T}_2$	$(\hat{T}_2 - T_1)/T_0$	Rank
FA	1.4E-01	3.05E-01	5.59E+02	3.99E+03	2
DFO	1.4E-01	3.05E-01	8.08E+02	5.77E+03	4
HHO	1.4E-01	3.05E-01	7.44E+02	5.31E+03	3
BOA	1.4E-01	3.05E-01	4.73E+02	3.38E+03	1

$$ {}^{t}Div = \frac{1}{N|L|}\sum_{i=1}^{N}\sqrt{\sum_{j=1}^{D}\left(x_i^j - \bar{x}^j\right)^2} \tag{10} $$

in which, ${}^{t}Div$ indicates the current diversity the population; N represents the population size; D shows the dimension of the problem; L specifies the longest diagonal length of search space. Also, x_i^j returns the jth component of the ith agent, and $\overline{x}^j$ is defined as the mean value of all jth components in the population.

When the diversity index is lower within a population, it signifies that the agents are concentrated in a smaller region of the search domain, emphasizing their search efforts in that specific location. As a result, a lower diversity index indicates that the algorithm's main search behavior is focused on exploration. Conversely, higher values of the diversity index indicate that the algorithm's primary search behavior is geared towards exploitation. Hence, the exploration and exploitation behaviors of the algorithm can be mathematically formulated using the normalized formulation provided below.

$$ \begin{aligned} {}^{t}NDiv\% &= \frac{{}^{t}Div}{{}^{t}Div_{max}} \times 100 \\ {}^{t}\overline{NDiv}\% &= \frac{{}^{t}Div_{max} - {}^{t}Div}{{}^{t}Div_{max}} \times 100 \end{aligned} \tag{11} $$

in which, ${}^{t}Div_{max}$ show the maximum diversity indexes during the optimization process. ${}^{t}NDiv\%$ and its conjugate (${}^{t}\overline{NDiv}\%$) values respectively represent the exploration and exploitation indexes. Based on this formulation the behavioral diagram for the Rastrigin function (as the selected function for assessing the behavior of the methods) is given in Fig 2. Based on the given diagrams all algorithms shift their behavior from exploration to exploitation. This indicates that all of them works proper (i.e., first explore the search domain and when the promising region of the search space is spotted they makes exploitation search). Among the all methods, the HHO is the most rapid algorithm in changing its behavior (from 100% exploration to 100% exploitation). Also, the BOA could not make a complete shift and it means that it nearly could use %95 of its expected search potential.

3.2. Sizing optimization of truss structures

To evaluate the performance of the selected method on the handling the complex constrained optimization problems, in this section their performances are tested on handling two different structural optimization problems. To provide a more challenging state, the decision variables are selected from both discrete and continuous search domains under multiple loading conditions. The termination criteria for stopping the process are either reaching 100,000 NSAs (to avoid premature termination) or encountering 50 null iterations (iterations without improvement), which one is achieved faster.

3.2.1. A 200 bar-dome structure

This section focuses on examining the problem of minimizing weight in the 200-bar planar truss structure described in Fig. 3. The truss comprises 29 separate groups, which are detailed in Table 10. The material used has a modulus of elasticity of 30 Msi and a density of 0.283 lb/in^3. The stress of both tensile and compressive members is limited to ±10.0 ksi, and there are no restrictions on displacements. The minimum acceptable cross-sectional area for the elements is 0.1 in2. The structure is designed to withstand three different loading conditions. In the first loading condition, nodes 1, 6, 15, 20, 29, 34, 43, 48, 57, and 62 experience a load of 1 kips in the *x* direction. For the second loading condition, nodes 1-6, 8, 10, 12, 14-20, 22, 24, 26, 28-34, 36, 38, 40, 42-48, 50, 52, 54, 56-62, 64, 66, 68, and 70-75 experience a load of 10 kips in the negative y direction. In the third loading condition, the first and second loading conditions are combined. The optimal results are documented in Table 11. Based on the given results the HHO could find the lightest structural system. Also, acquired standard deviation (Std.) values indicate that HHO shows the most stable behavior among all methods. The BOA and the HHO methods require nearly the same amount of computational cost, while it is lower than the other selected methods.

3.2.2. A 354-bar dome structure

The next example in this study focuses on a 354-bar braced dome structure, illustrated in Fig. 4. This structure serves as the roof system for a large hall, with a span of 131.23 ft (40m) and a height of 27.17 ft (8.28m). It consists of 354 members and 127 nodes, as depicted in Fig. 4. The sizing variables for this example are chosen from a discrete set sourced from the AISC database. According to the ASCE7-05 specification, this structure is subjected to three main load combinations: (i) $D + S$, (ii) $D + S + W$ for negative pressure cases, and (iii) $D + S + W$ for positive pressure cases. Here, D, S, and W refer to the dead, live, and wind loads, respectively.

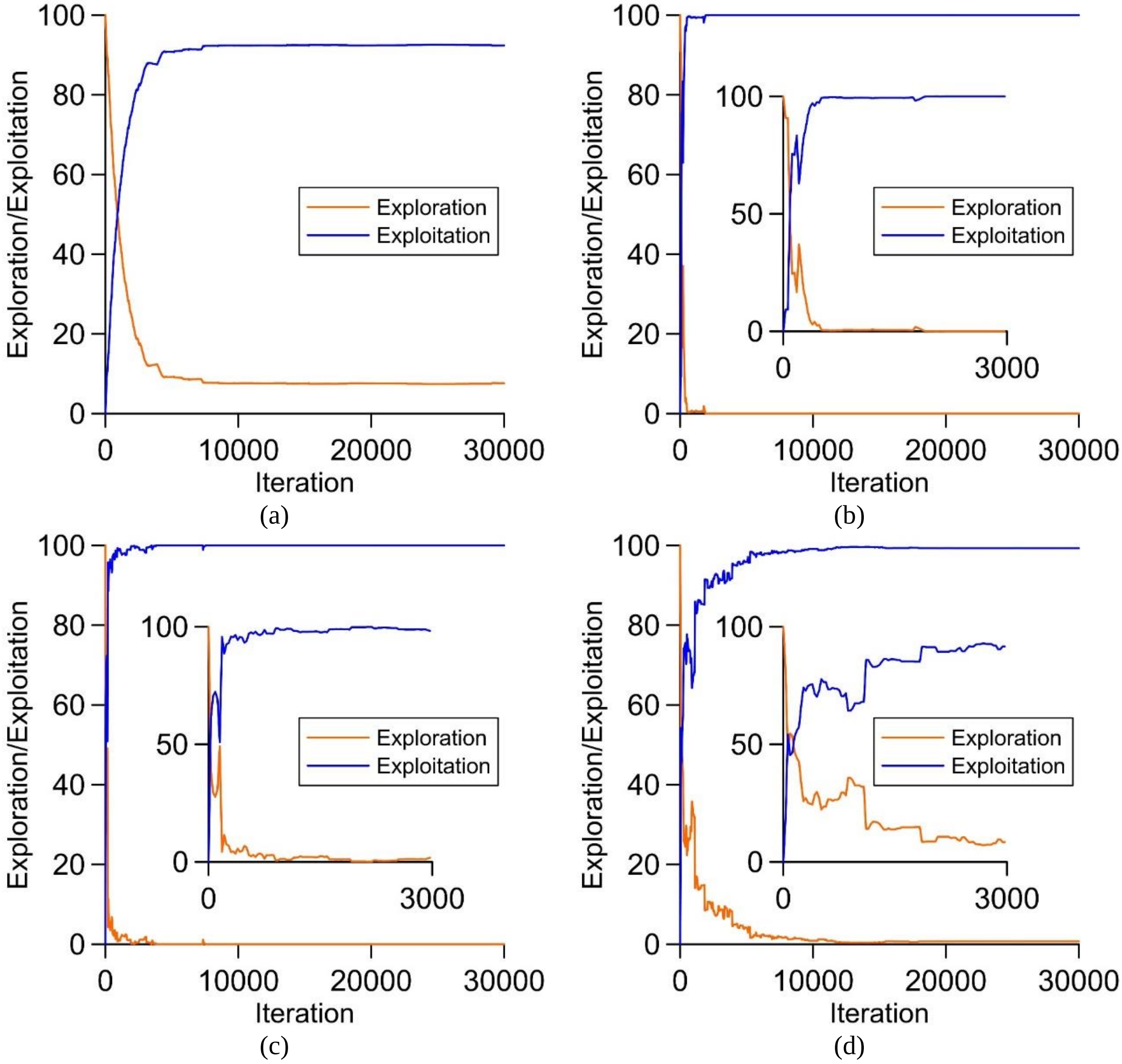


Fig. 2. Behavioral diagram for Rastrigin function for (a) BOA, (b) HHO, (c) DFO and (d) FA methods

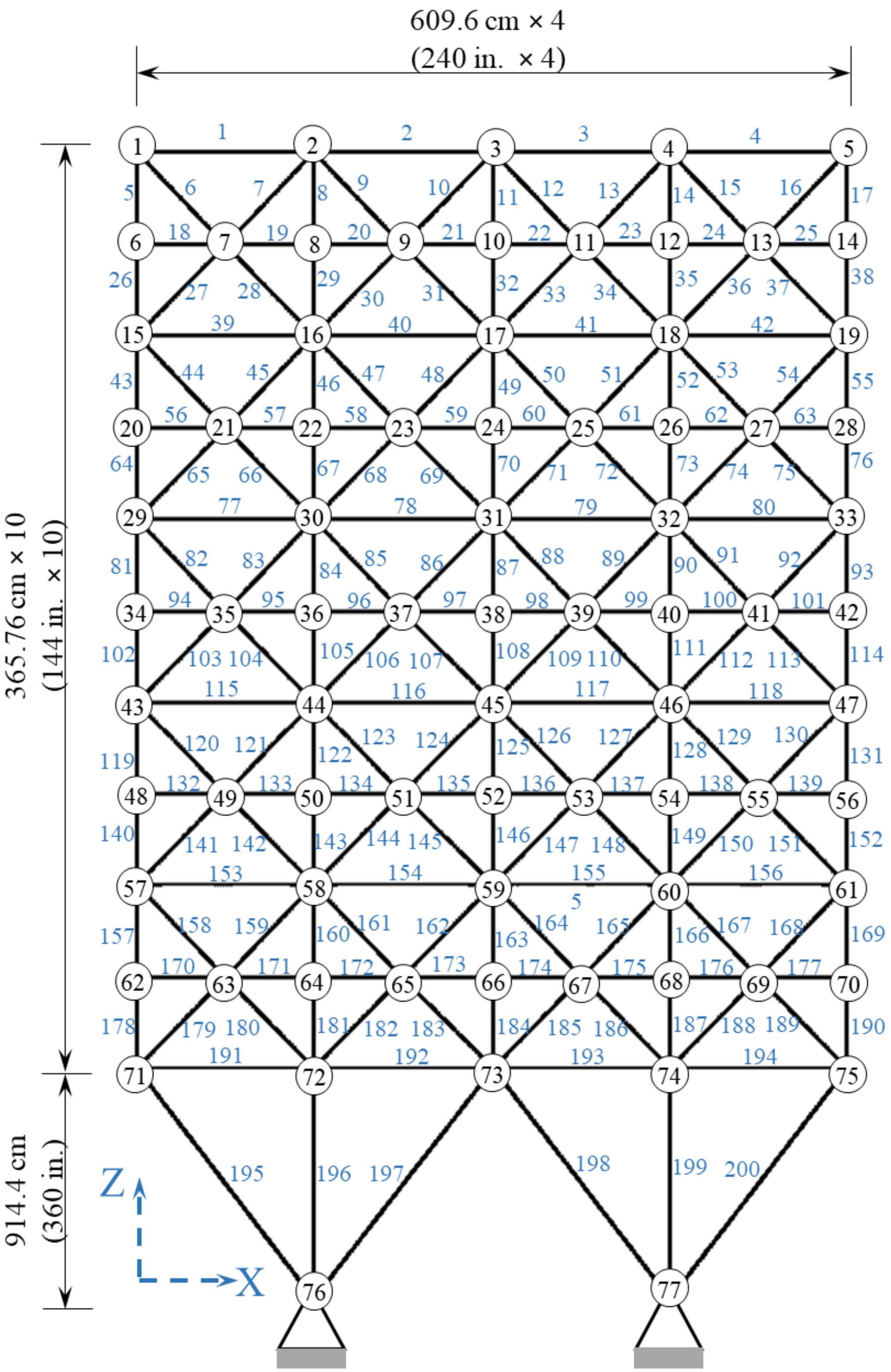


Fig. 3. The 200-bar planar truss structure

Table 10. Grouping of 200-bar planar truss structure

Group	Group members
1	1-2-3-4
2	5-8-11-14-17
3	19-20-21-22-23-24
4	18-25-56-63-94-101-132-139-170-177
5	26-29-32-35-38
6	6-7-9-10-12-13-15-16-27-28-30-31-33-34-36-37
7	39-40-41-42
8	43-46-49-52-55
9	57-58-59-60-61-62
10	64-67-70-73-76
11	44-45-47-48-50-51-53-54-65-66-68-69-71-72-74-75
12	77-78-79-80
13	81-84-87-90-93
14	95-96-97-98-99-100
15	102-105-108-111-114
16	82-83-85-86-88-89-91-92-103-104-106-107-109-110-112-113
17	115-116-117-118
18	119-122-125-128-131
19	133-134-135-136-137-138
20	140-143-146-149-152
21	120-121-123-124-126-127-129-130-141-142-144-145-147-148-150-151
22	153-154-155-156
23	157-160-163-166-169
24	171-172-173-174-175-176
25	178-181-184-187-190
26	158-159-161-162-164-165-167-168-179-180-182-183-185-186-188-189
27	191-192-193-194
28	195-197-198-200
29	196-199

To simplify the computational process, the asymmetrical condition for snow load is disregarded. The corresponding load cases are shown in Fig. 5. The wind load is simulated on the curved surface, while the snow load cases are calculated on the projected area of the dome. The dome members are constructed using sandwich-type aluminum cladding, resulting in a dead load (pressure) of 0.2 kN/m^2 over the frame network of the structure. The stability and stress criteria adhere to the specifications of the AISD-ASD code [44].

Displacement is limited to 4.37 in (11.1 cm) in any direction for all structure nodes. Table 12 provides the results obtained by all selected methods. Based on the given results, for the terms of stability and accuracy the HHO outperforms the other methods by finding the lightest structural system and lowest standard deviation(Std.) value, respectively; However, the BOA stands in the first place from computational point of view (please note that this method could find near optimal solution compared with HHO).

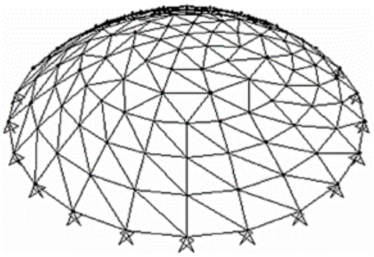
3.2.3. A 582-bar truss tower

The final case study focuses on a structural optimization example, specifically the weight minimization of a spatial 582-bar tower depicted in Fig. 6. To preserve symmetry, the structure's members are divided into 32 distinct groups. The tower is subjected to three load conditions, which are as follows:

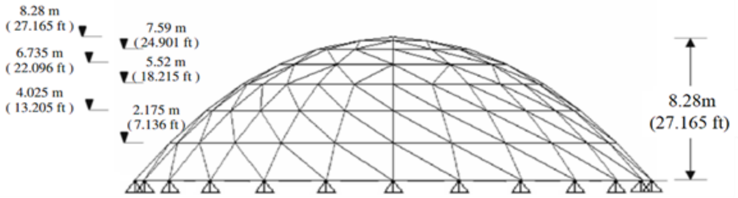
I. A vertical load of -6.75 kips on each node.
II. A horizontal load of 1.12 kips is applied on each node in the *x*-direction.
III. A horizontal load of 1.12 kips is applied on each node in the *y*-direction.

Table 11. Comparison of optimal designs for the 200-bar planar truss structure

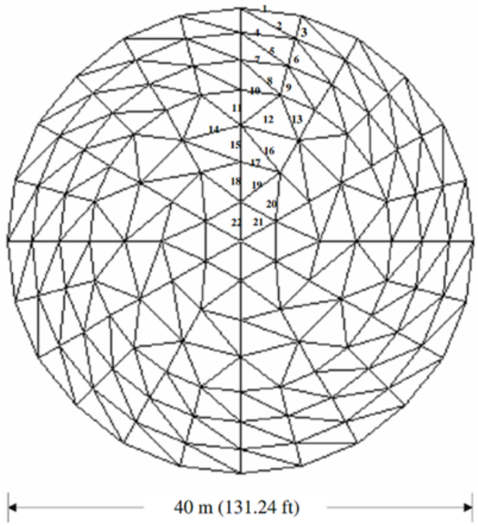
Design Variables	Optimal cross –sectional area (in^2)			
	FA	DFO	HHO	BO
A_1	0.1425	0.1144	0.1540	0.7590
A_2	0.9637	0.9443	0.9410	0.9032
A_3	0.1005	0.1310	0.1000	1.1000
A_4	0.1000	0.1016	0.1000	0.9952
A_5	1.9514	2.0353	1.9420	2.1350
A_6	0.2957	0.3126	0.3010	0.4193
A_7	0.1156	0.1679	0.1000	1.0041
A_8	3.1133	3.1541	3.1080	2.8052
A_9	0.1006	0.1003	0.1000	1.0344
A_{10}	4.1100	4.1005	4.1060	3.7842
A_{11}	0.4165	0.4350	0.4090	0.5269
A_{12}	0.1843	0.1148	0.1910	0.4302
A_{13}	5.4567	5.3823	5.4280	5.2683
A_{14}	0.1000	0.1607	0.1000	0.9685
A_{15}	6.4559	6.4152	6.4270	6.0473
A_{16}	0.5800	0.5629	0.5810	0.7825
A_{17}	0.1547	0.4010	0.1510	0.5920
A_{18}	8.0132	7.9735	7.9730	8.1858
A_{19}	0.1000	0.1092	0.1000	1.0362
A_{20}	9.0135	9.0155	8.9740	9.2062
A_{21}	0.7391	0.8628	0.7190	1.4774
A_{22}	0.7870	0.2220	0.4220	1.8336
A_{23}	11.1795	11.0254	10.8920	10.611
A_{24}	0.1462	0.1397	0.1000	0.9851
A_{25}	12.1799	12.0340	11.8870	12.509
A_{26}	1.3424	1.0043	1.0400	1.9755
A_{27}	5.4844	6.5762	6.6460	4.5149
A_{28}	10.1372	10.7265	10.8040	9.8000
A_{29}	14.5262	13.9666	13.8700	14.5310
Best Weight (lb)	25,521.81	25,674.83	25,491.9	28,537.8
Mean Weight (lb)	31,264.22	30,809.8	28,550.93	31,676.96
NSAs	20,760	27,030	15,090	15,300
Std. (lb)	1814.55	1426.78	1010.51	1580.63



a) 3D view



b) Side view



c) Top view

Fig. 4. Schematic for 354-bar truss dome structure.

Table 12. Comparison of optimization results for 354-bar braced dome

Sizing variables	Optimal cross-sectional areas (in.^{2})			
	FA	DFO	HHO	BOA
1	1.07	1.07	1.07	1.48
2	3.17	3.17	3.17	2.68
3	2.23	2.23	2.23	4.3
4	2.68	2.68	2.68	2.68
5	2.23	2.23	2.23	2.23
6	2.23	2.23	2.23	2.23
7	2.23	2.23	2.23	2.23
8	2.23	2.23	2.23	2.23
9	1.7	1.7	1.7	1.7
10	2.23	2.23	2.23	2.25
11	1.7	2.66	1.7	2.66
12	1.7	2.23	1.7	2.23
13	1.7	1.7	1.7	2.23
14	1.7	1.7	1.7	1.7
15	1.7	1.7	1.7	2.25
16	1.7	1.7	1.7	1.7
17	1.48	1.48	1.48	1.7

Table 12. Continued

18	1.48	2.68	1.48	3.17
19	1.07	1.07	1.07	1.48
20	1.07	1.07	1.07	1.48
21	1.07	1.07	1.07	1.48
22	1.07	1.48	1.07	1.7
Best weight (lb)	32,574.9	33,557.5	32,574.9	36,343.3
Mean weight (lb)	34,165.8	34,003.8	32,680.2	38,845.3
NSAs	35,925	37,800	12,575	11,550
Std.	3,489.2	1,845.1	1,835.8	1,985.8

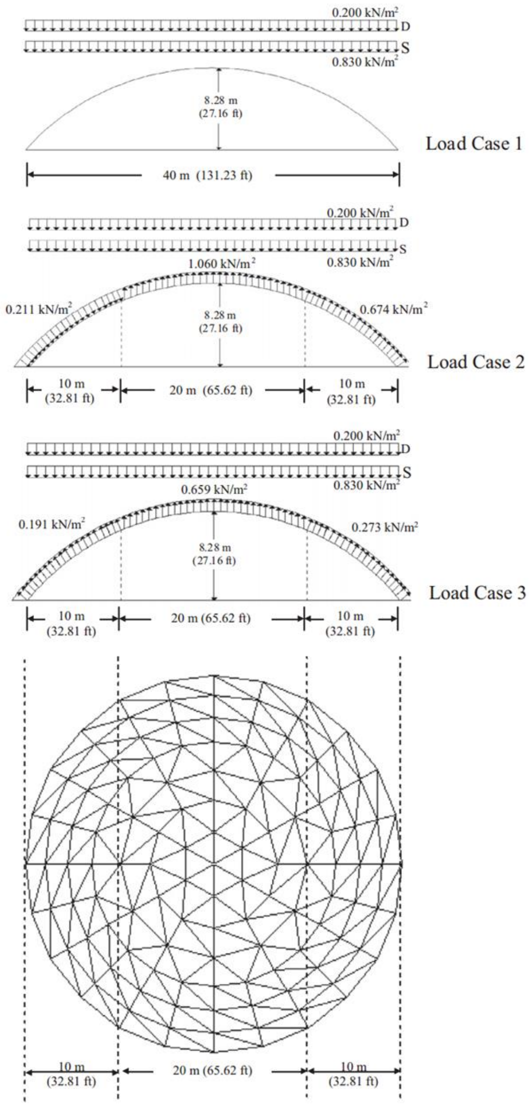


Fig. 5. The three load cases for 354-bar truss dome structure

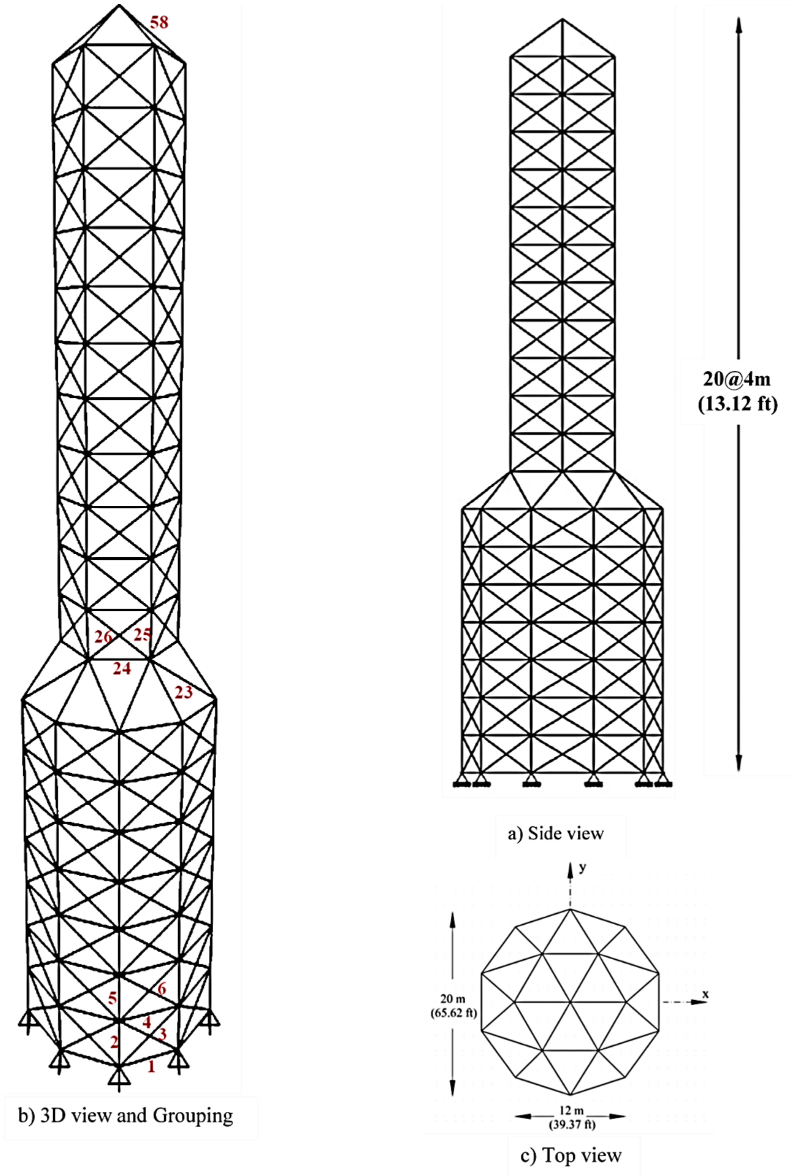


Fig. 6. The 582-bar truss tower

The sizing variables for this example have been chosen from the discrete set outlined in Table 13. The members of this set comprise 140 W-shape profiles provided in the steel structural profiles of AISC-ASD.

The upper and lower bounds for these variables are restricted to 6.16 in^2 (39.74 cm^2) and 215.00 in^2 (1387.09 cm^2) respectively. The nodal displacement for all principal directions is constrained to a maximum of 3.15 in (8 cm). The stress limitation is determined according to the buckling criterion specified in the AISD-ASD89 code as follows [31]:

$$\begin{cases} \sigma_i^+ = 0.6F_y & \sigma_i \geq 0 \\ \sigma_i^- & \sigma_i < 0 \end{cases} \tag{12}$$

Where F_y represents the yielding stress of the materials and σ_i^- and σ_i^+ respectively indicate the compressive and tensile stresses. To impose the buckling criteria, the compressive stress is determined based on the slenderness ratio of the tensile member as follows:

$$\sigma_i^- = \begin{cases} \left[\left(1-\frac{\lambda_i^2}{2C_c^2}\right)F_y \Big/ \left(\frac{5}{3}+\frac{3\lambda_i}{8C_c}-\frac{\lambda_i^3}{8C_c^3}\right)\right] & for\ \lambda_i < C_c \\ \frac{12\pi^2 E}{23\lambda_i^2} & for\ \lambda_i \geq C_c \end{cases} \tag{13}$$

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

$$C_c = \sqrt{\frac{2\pi^2 E}{F_y}} \tag{14}$$

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:

$$\lambda_i = \frac{k_i l_i}{r_i} \leq \begin{cases} 300\ for\ tension\ members \\ 200\ for\ compression\ members \end{cases} \tag{15}$$

where λ_i, r_i, and l_i determine the *i*th member's slenderness ratio, radius of gyration and length, respectively. For compression elements, under any condition the allowable stress must not surpass the value of $12\pi^2 E/23\lambda_i^2$ ever [31]. In this particular case, the complex structural optimization problem is tackled using four selected techniques, and the obtained results are compiled and presented in Table 14. For each algorithm, the number of structural analyses (NSAs) conducted and the corresponding standard deviation (Std.) values are provided. Based on the given results the HHO can outperform the other methods in the terms of stability (with lowest standard deviation value) and structure's weight. However, the BOA can reach to optimal solution (by finding a local optimum) in the less NSAs value compared with all other methods.

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

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

Groups	Optimal cross-sectional areas			
	FA	DFO	HHO	BO
1	W8×21	W8×24	W8×21	W8×24
2	W24×76	W12×72	W24×84	W24×68
3	W8×21	W8×28	W8×21	W8×28
4	W12×65	W12×58	W24×62	W18×60
5	W8×21	W8×24	W8×21	W8×24
6	W8×21	W8×24	W8×21	W8×24
7	W10×54	W10×49	W16×57	W21×48
8	W8×21	W8×24	W8×21	W8×24
9	W8×21	W8×24	W8×21	W10×26
10	W12×50	W12×40	W12×53	W14×38
11	W8×21	W12×30	W8×21	W12×30
12	W10×68	W12×72	W10×77	W12×72
13	W24×76	W18×76	W21×83	W21×73
14	W14×53	W10×49	W21×57	W14×53
15	W12×79	W14×82	W18×76	W18×86
16	W8×21	W8×31	W8×21	W8×31
17	W12×65	W14×61	W10×22	W18×60
18	W8×21	W8×24	W18×55	W8×24
19	W8×21	W8×21	W8×21	W16×36
20	W12×45	W12×40	W8×21	W10×39
21	W8×21	W8×24	W14×30	W8×24
22	W8×21	W14×22	W8×21	W8×24
23	W16×26	W8×31	W8×21	W8×31
24	W8×21	W8×28	W8×21	W8×28
25	W8×21	W8×21	W8×21	W8×21
26	W8×21	W8×21	W8×21	W8×24
27	W8×21	W8×24	W10×22	W8×28
28	W8×21	W8×28	W8×21	W14×22
29	W8×21	W16×36	W8×21	W8×24
30	W8×21	W8×24	W8×31	W8×24
31	W8×21	W8×21	W8×21	W14×22
32	W8×21	W8×24	W12×22	W8×24
Best Volume (m^3)	20.3	22.07	20.07	22.37
Mean Volume (m^3)	20.87	22.75	20.53	22.62
NSAs	25890	24050	17670	8880
Std. Dev. (m^3)	0.53	0.51	0.22	0.32

4. Conclusion

The present study focuses on conducting a comparative analysis of four swarm-inspired metaheuristic algorithms on solving sizing optimization of the truss systems. The selected algorithms, arranged chronologically based on their emergence dates, are the Firefly Algorithm (FA), Drosophila Food-Search Optimization (DFO), Harris Hawk Optimization (HHO), and Butterfly Optimization Algorithm (BOA). To provide a comprehensive overview of the selected algorithms, they were initially compared based on their distinct attributes using various mathematical benchmark functions (with continuous search domains and no constraints). Subsequently, their performance was specifically evaluated in the context of sizing optimization for truss structures. These cases not only encompass a variety of constraints search spaces but also involve discrete and continuous design variables. The methods are evaluated and compared based on various criteria, including search behavior, stability, complexity, convergence rate, and accuracy level. This comprehensive assessment provides insights into the strengths and weaknesses of each algorithm across different problem types.

The achieved outcomes are provided through the illustrative tables and diagrams. Obtained numeric result show that nearly for all cases the HHO could outperform all other algorithms on the term of accuracy. This means that in nearly all cases this method can find the most optimal value according to the objective function of the problem. The behavioral diagrams indicate that expect the BOA the other algorithms show full shifting behavior (i.e., during the optimization process their search behavior is completely shifted from full exploration to full exploitation). Computed complexity analysis reveals that, the BOA has simplest search algorithm while the DFO has the most complex search algorithm among all other selected methods. Also, the stability tests (based on the standard deviation values for multiple runs) declare that the HHO, especially for sizing optimization problems, shows the most stable behavior during the entire optimization process. In terms of computational cost, the HHO leaves it gave up its throne to the BOA method, since the BOA generally required the lowest number of objective function evaluation for both mathematical and structural problems. Consequently, based on all addressed facts, especially for sizing optimization of truss systems, the HHO can be recommended as an effective search technique compared the other selected methods.

Conflict of interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

This research received no external funding.

Data availability statement

No new data were created or analyzed in this study.

References

[1] Rade J, Balu A, Herron E, Pathak J, Ranade R, Sarkar S, Krishnamurthy (2021) Algorithmically consistent deep learning frameworks for structural topology optimization. Engineering Applications of Artificial Intelligence106:104483.

[2] Mortazavi A (2019) The performance comparison of three metaheuristic algorithms on the size, layout and topology optimization of truss structures. Muğla Journal of Science and Technology 5:28-41.

[3] Mortazavi A (2019) Interactive fuzzy search algorithm: A new self-adaptive hybrid optimization algorithm. Engineering Applications of Artificial Intelligence 81:270-82.

[4] Mortazavi A (2022) Interactive fuzzy Bayesian search algorithm: a new reinforced swarm intelligence tested on engineering and mathematical optimization problems. Expert Systems with Applications 187:115954.

[5] Mortazavi A, Moloodpoor M (2021) Differential evolution method integrated with a fuzzy decision-making mechanism and Virtual Mutant agent: Theory and application. Applied Soft Computing 112:107808.

[6] Moloodpoor M, Mortazavi A, Ozbalta N (2021) Thermo-economic optimization of double pipe heat exchanger using a compound swarm intelligence. Heat Transfer Research.

[7] Mafarja M, Aljarah I, Heidari AA, Hammouri AI, Faris H, Al-Zoubi AM, Mirjalili S (2018) Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems. Knowledge-Based Systems 145:25-45.

[8] Souza RRd, Fadel Miguel LF, Lopez RH, Miguel LFF, Torii AJ (2016) A procedure for the size, shape and topology optimization of transmission line tower structures. Engineering Structures 111:162-84.
[9] Mortazavi A, Toğan V (2017) Triangular units-based method for simultaneous optimizations of planar trusses. Advances in Computational Design 2:195-210.
[10] Mortazavi A (2021) Solving structural optimization problems with discrete variables using interactive fuzzy search algorithm. Structural Engineering and Mechanics 79:247-65.
[11] Holland JH (1984) Genetic Algorithms and Adaptation. In: Selfridge OG, Rissland EL, Arbib MA (ed) Adaptive Control of Ill-Defined Systems Boston, MA: Springer US, Inc., pp. 317-33.
[12] Rechenberg I. Evolutionsstrategien. In: Schneider B, Ranft U (ed) Simulations methoden in der Medizin und Biologie Berlin, Heidelberg. Springer Berlin Heidelberg; 1978. Inc.,pp. 83-114.
[13] Javidy B, Hatamlou A, Mirjalili S (2015) Ions' motion algorithm for solving optimization problems. Applied Soft Computing 32:72-9.
[14] Patel VK, Savsani VJ (2015) Heat transfer search (HTS): a novel optimization algorithm. Information Sciences 324:217-46.
[15] Glover F (1989) Tabu search-part i. ORSA Journal on Computing 1:190-206.
[16] Rao RV, Savsani VJ, Vakharia DP (2011) Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems. Computer-Aided Design 43:303-15.
[17] Kennedy J, Eberhart R (1995) Particle swarm optimization. Neural Networks. In: Proceedings, IEEE International Conference on1995. vol.4, pp. 1942-1948.
[18] Dorigo M, Maniezzo V, Colorni A (1996) Ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 26:29-41.
[19] Yang XS (2019) Firefly Algorithms for Multimodal Optimization. In: Watanabe O, Zeugmann T (ed) Stochastic Algorithms: Foundations and Applications. Berlin, Heidelberg: Springer Berlin Heidelberg, Inc., pp. 169-78.
[20] Cheng MY (2014) Prayogo D. Symbiotic Organisms Search: A new metaheuristic optimization algorithm. Computers & Structures 139:98-112.
[21] Das KN, Singh TK (2014) Drosophila food-search optimization. Applied Mathematics and Computation 231:566-80.
[22] Liang YC, Cuevas Juarez JR (2016) A novel metaheuristic for continuous optimization problems: Virus optimization algorithm. Engineering Optimization.48:73-93.
[23] Arora S, Singh S (2019) Butterfly optimization algorithm: a novel approach for global optimization. Soft Computing 23:715-34.
[24] Mortazavi A, Togan V (2021) Metaheuristic algorithms for optimal design of truss structures. In: Nigdeli SM, Bekdaş G, Kayabekir AE, Yucel M (ed) Advances in Structural Engineering-Optimization: Emerging Trends in Structural Optimization Cham: Springer International Publishing, Inc., pp. 199-220.
[25] Mortazavi A, Togan V (2021) Solving of Distinct Engineering Optimization Problems Using Metaheuristic Algorithms. In: Carbas S, Toktas A, Ustun D(ed) Nature-Inspired Metaheuristic Algorithms for Engineering Optimization Applications Singapore: Springer Singapore,Inc., pp. 111-36.
[26] Mortazavi A (2019) Comparative assessment of five metaheuristic methods on distinct problems. Dicle University Journal of Engineering 10:879.
[27] Mortazavi A, Toğan V, Daloğlu A, Nuhoğlu A (2018) Comparison of two metaheuristic algorithms on sizing and topology optimization of trusses and mathematical functions. Gazi University Journal of Science 31:416-35.
[28] Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen HH (2019) Hawks optimization: Algorithm and applications. Future Generation Computer Systems 97:849-72.
[29] Suganthan PN, Hansen N, Liang JJ, Deb K, Chen YP, Auger A, Tiwari S (2005) Problem Definitions and Evaluation Criteria for The CEC 2005 Special Session on Real-Parameter Optimization. Technical Report, Nanyang Technological University, Singapore and KanGAL Report 2005005, IIT Kanpur, India.
[30] Mortazavi A (2022) Interactive fuzzy Bayesian search algorithm: A new reinforced swarm intelligence tested on engineering and mathematical optimization problems. Expert Systems with Applications 187:115954.
[31] AISC (1989) American Institute of Steel Construction. Manual of steel construction allowable stress design. 9th ed. Chicago, IL.