Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 6078986, 13 pages
https://doi.org/10.1155/2022/6078986

Research Article

A Comparative State-of-the-Art Constrained Metaheuristics Framework for TRUSS Optimisation on Shape and Sizing

Bahareh Etaati,[1] **Amin Abdollahi Dehkordi**,[2] **Ali Sadollah**,[3] **Mohammed El-Abd**,[4] **and Mehdi Neshat**[5]

[1]*Amirkabir University of Technology, Department of Computer Engineering and Information Technology, Tehran, Iran*
[2]*Computer Engineering Department, Najafabad Branch, Islamic Azad University, Najafabad, Iran*
[3]*Department of Mechanical Engineering, University of Science and Culture, Tehran, Iran*
[4]*College of Engineering and Applied Sciences, American University of Kuwait, Salmiya, Kuwait*
[5]*Center for Artificial Intelligence Research and Optimisation, Torrens University Australia, Brisbane, Australia*

Correspondence should be addressed to Bahareh Etaati; b.etaati91@aut.ac.ir

Received 4 February 2022; Revised 27 February 2022; Accepted 28 February 2022; Published 26 March 2022

Academic Editor: Man Fai Leung

In order to develop the dynamic effectiveness of the structures such as trusses, the application of optimisation methods plays a significant role in improving the shape and size of elements. However, conjoining two heterogeneous variables, nodal coordinates and cross-sectional elements, makes a challenging optimisation problem that is nonlinear, multimodal, large-scale with dynamic constraints. To handle these challenges, evolutionary and swarm optimisation algorithms can be robust and practical tools and show great potential to solve such complex problems. This paper proposed a comparative truss optimisation framework to solve two large-scale structures, including 314-bar and 260-bar trusses. The proposed framework consists of twelve state-of-the-art bio-inspired algorithms. The experimental results show that the marine predators algorithm (MPA) performed best compared with other algorithms in terms of convergence speed and the quality of the proposed designs of the trusses.

1. Introduction

The dynamic performance of structures exposed to various dynamic loading is connected with their fundamental natural frequencies. For instance, prior knowledge of the natural frequencies of a structure may help prevent the vibration and noise produced under dynamic loadings, such as wind or earthquake. As a result, obtaining the optimal sizing and layout of structures with frequency constraints is exceptionally important to enhance the dynamic behaviour of structures [1].

Truss optimisation has been attracting many researchers over the past decades as one of the most significant subjects in structural engineering. Design variables include the truss sizing, shape, and topology, and the main optimisation problems include the optimisation of the design variables. In most of the case studies, the size of bars comes from a set of discrete values; therefore, the applications of the discrete optimisation methods are considerable (for further study on the discrete optimisation methods see [2]). Most studies were conducted to obtain the optimal set of sizing variables in order to minimize the structural weight. However, the optimal structural weight depends on different design variables rather than just one. For example, the optimal truss shape is affected by its topology and size and vice versa. With this in mind, the simultaneous optimisation of design variables with frequency constraints has attracted many researchers recently.

Nevertheless, coupling shape and sizing variables may lead to mathematical difficulties, nonoptimal solutions, and occasionally divergence problems. Additionally, frequency constraints are extremely nonlinear, nonconvex, and implicit regarding design variables [3]. Therefore, global optimisation algorithms, which are able to find the global best solution in the search space, could be a good solution to truss shape and sizing optimisation with frequency constraints.

Overall, two types of optimisation methods are applied to truss optimisation problems, namely mathematical programming techniques and meta-heuristic algorithms. While mathematical programming techniques have a fast convergence speed, they are complicated and time-consuming due to the necessity of sensitivity analysis. Also, they are dependent on the starting structural design and prone to being trapped in local minima. Owing to these drawbacks of the mathematical programming techniques, meta-heuristic algorithms have been mostly used for structural optimisation as effective global optimisation methods. These stochastic search methods are bio-inspired, easy to implement, problem-independent, and flexible. Additionally, they have strong exploration and exploitation abilities, which makes them free from having prior gradient knowledge of the objective function and being sensitive to the initial point.

As already mentioned, meta-heuristic approaches inspired by biological processes are a large class of global optimisation techniques that have attracted many studies in the subject of truss optimisation [4]. Swarm intelligence-based methods, inspired by living organisms' social behaviour, are a group of population-based meta-heuristics. These algorithms proved to be great optimizers for truss problems in recent years [5–10].

In this paper, twelve modern swarm optimisation methods are deployed for the shape and sizing optimisation of a large-scale truss problem with frequency constraints. These state-of-the-art algorithms include grey wolf optimizer (GWO) [11], moth flame optimizer (MFO) [12], multi-verse optimizer (MVO) [13], dragonfly algorithm (DA) [14], equilibrium optimizer (EO) [15], arithmetic optimisation algorithm (AOA) [16], Generalized Normal Distribution optimisation (GNDO) [17], Salp Swarm Algorithm (SSA) [18], Marine Predator Algorithm (MPA) [19], Henry Gas Solubility optimisation (HGSO) [20], Neural Network Algorithm (NNA) [21], and Water Cycle Algorithm (WCA) [22]. GWO simulates grey wolves' leadership hierarchy and hunting behavior in nature, MFO is inspired by moths' navigation behavior, MVO is based on white hole, black hole, and wormhole concepts in cosmology, DA is based of searching behavior of dragonflies in static and dynamic swarms, EO is inspired by control volume mass balance models applied for dynamic and equilibrium states' estimation, AOA is based on the distribution behavior of arithmetic operators in mathematics, GNDO uses generalized normal distribution models to update the population, SSA is inspired by salps' swarming behavior during navigation and foraging in oceans, MPA is motivated by foraging strategy and optimal rate policy between prey and predator in oceans, and HGSO simulates the Henry's law behavior. All the algorithms are compared in terms of the optimal weight of truss structures concerning frequency constraints. In short, the main contributions of the paper are as follows:

(1) Evaluating two large-scale truss problems (314-bar and 260-bar) with frequency constrains in order to optimise the shape and sizing variables

(2) A comparison of twelve state-of-the-art bio-inspired optimisation methods

(3) Tuning the population size of the best-performed optimisation method

The rest of the paper is organized as follows: Section 2 provides a brief review of meta-heuristic algorithms recently proposed for truss optimisation problems. Section 3 describes two truss problem case studies in more detail. Section 4 explains the optimisation methods applied to the truss problem. The given experimental results of the methods' performance on the case studies are represented in Section 5. Lastly, Section 6 concludes and provides a couple more beneficial suggestions for future work.

2. Related Work

This section focuses on recently proposed meta-heuristic algorithms for truss optimisation problems. Rahami et al. [23] employed a genetic algorithm (GA) coupled with a force method for truss sizing, shape, and topology optimisation. This study aimed to decrease the number of input variables, increasing the GA's convergence speed and reducing its computational cost. In another work, Wei et al. [1] proposed a Niche Hybrid Parallel Genetic Algorithm (NHPGA) to optimise truss shape and sizing. The authors combined GA, parallel computing, and simplex search with a niche approach in order to speed up the GA's search ability to find optimal solutions. An improved differential evolution (ReDE) was introduced in [24] for structural shape and size optimisation that used the roulette wheel selection technique to improve the efficiency of the basic DE.

Additionally, a novel hybrid DE and symbiotic organism search algorithm (SOS) was proposed in [25] to optimise the shape and sizing of truss structures. The algorithm utilised the global searching ability of DE and the local searching ability of SOS to achieve optimal solutions. Lamberti [26] proposed a multi-level population-based simulated annealing algorithm called CMLPSA for sizing and shape optimisation. Azad [27] hybridised an adaptive dimensional search with two versions of the big bang-big crunch algorithm for sizing optimisation of truss structures. In [28], a mine blast algorithm was proposed for truss sizing optimisation. The algorithm was inspired by the explosion of mine bombs in the real world.

The most popular swarm optimisation method applied to this kind of problem is particle swarm optimisation (PSO) [5] inspired by the birds flock or fish school's social behaviour. As an example, in [6], Kaveh et al. proposed a democratic PSO (DPSO) for truss sizing and topology optimisation with frequency constraints. This research aimed to alleviate the basic PSO premature convergence in frequency constraints. Other classical swarm methods applied for truss optimisation include ant colony optimisation (ACO) [7], artificial bee colony (ABC) algorithm [29, 30], and shuffled frog leaping (SFL) [31] to name but a few. More recently, new modern swarm optimisation algorithms have been proposed and developed for structural design optimisation. For example, in [32], a new swarm method named ray optimisation, motivated by Snell's law of light refraction, was proposed for truss optimisation. Then, in [33], the

authors proposed an improved ray optimisation method to optimise the sizing and topology of truss structures. Other new swarm-intelligence methods applied to truss problems include firefly algorithm [8, 34], dolphin echolocation [9, 35], teaching-learning based optimisation [36], grey wolf optimizer [10], political optimizer [37], and imperialist competitive algorithm [38, 39], to name but a few.

3. Truss Problem Formulation

Overall, in a truss sizing and shape optimisation problem with multiple frequency constraints, the primary purpose is to minimize the truss weight while meeting the constraints on natural frequencies. Nodal coordinates and cross-sectional areas are design variable, which change persistently during design process. In this problem, the truss topology is supposed to be unalterable and predetermined. Additionally, design variables may be limited to a specific interval. Hence, the optimisation problem can be stated as follows:

\begin{align}
&\text{Find}: & X &= \{A, C\} \nonumber
&\text{Where}: & A &= \{A_{1}, A_{2},\ldots, A_{n}\}\nonumber
& & C &= \{C_{1},C_{2},\ldots,C_{m}\} \nonumber
&\text{Minimize}:& f(X) &= \sum_{i=1}^{nm} \rho_{i} A_{i} L_{i}
&\text{Subjected}:& \omega_{j} & \geq \omega_{j}^{Lb} \nonumber
& & \omega_{k} & \leq \omega_{k}^{Ub}\nonumber
& & A^{Lb} & \leq A \leq A^{Ub}\nonumber
& & C^{Lb} & \leq C \leq C^{Ub}\nonumber
\nonumber.
\end{align}

Here, X vector includes both cross-sectional areas and the nodal coordinates of the structure. Ci is nodal coordinates of the ith node of the structure. $f(X)$ is the structural weight, and ρi, Ai, as well as Li are the material density, cross-sectional area, and length of ith element, respectively. Also, n and m represent the number of structural cross-sectional areas and nodal coordinates confined to lower and upper bounds $[A^{Lb}, A^{Ub}]$ and $[C^{Lb}, C^{Ub}]$, respectively. Additionally, ωj and ω_k denote jth and kth natural frequencies restricted to the lower and upper bounds ω_j^{Lb} and ω_j^{Ub} [1].

In order to convert the constrained problem into an unconstrained one, we employed the function below:

\begin{equation}
f_{penalty}(X) = f(X) + (nTotalConstVio \times PF), \qquad nTotalConstVio = \sum_{i=1}^{k} (C_{vio_{i}}+ A_{vio_{i}})
\end{equation}

where $f_{\text{penalty}}(X)$ is the new objective function, n Total Const Vio and PF are the total amount of constraint violations, and the penalty factor, respectively. Also, vio_i is the violation value for ith constraint, which is set to zero for satisfied constraints. Here, the constraints include nodal displacement and element stress constraints.

Also, different values of the penalty factor PF were tested to tune this parameter. Finally, PF was set to 1000. The penalty function used is mostly similar to static penalties proposed in [40].

3.1. Case Study. In this paper, we aim to optimise two large-scale structural design problems proposed by the International Student Competition in Structural optimisation (ISCSO) in 2018 and 2019 [41], respectively. In the following subsection, the truss problems are explained in more detail.

3.1.1. 314-Bar Truss. The 314-bar problem introduced in [42] is a large-scale truss structure, and the main goal is to minimize the weight of a truss structure, including 314 bars and 84 nodes, according to the given constraints. The challenge is to obtain the optimal truss sizing and shape, while its topology is considered to be unalterable. Thus, the design variables include 314 sizing (A) and 14 shape (C) variables (treated as discrete decision variables). Figure 1 shows this truss structure in more detail.

The optimal solution is to minimize the structural weight, where no nodes and members violate the stress and displacement constraints. It means that the feasible solution possesses a value of zero for the amount of constraint violation, which is given by the function below:

\begin{equation}
[W,S_{Vio},D_{Vio}] = objective_function(A,C).
\end{equation}

Here, the function inputs include A and C denoting the cross-sectional areas and nodal coordinates, respectively. The outputs include the structural weight, the amount of stress and displacement violations, which are represented with W, DV io and SV io, respectively. Moreover, the sizing and shape variables can take only integer values limited to the lower and upper bounds [1, 37] and [9000, 20000], respectively.

3.1.2. 260-Bar Truss. The 260-bar problem introduced in [42] is to minimize the weight of a large-scale steel truss, consisting of 260 bars and 76 nodes, while satisfying the stress and displacement constraints. Also, the number of sizing (A) and shape (C) variables are considered 260 and 10, respectively. Additionally, the shape variables can take only integer values in [−25000, 3500]. The rest of the problem is similar to the ISCSO 2018. Figure 2 shows this truss problem in more detail.

4. Methodology

Optimising both structural parameters, sizing and shape based on the mass to considerable frequency constraints is challenging because the nature of these problems is highly nonlinear, multimodal, discrete and complex. Another difficulty of the truss optimisation problems is the mixing of

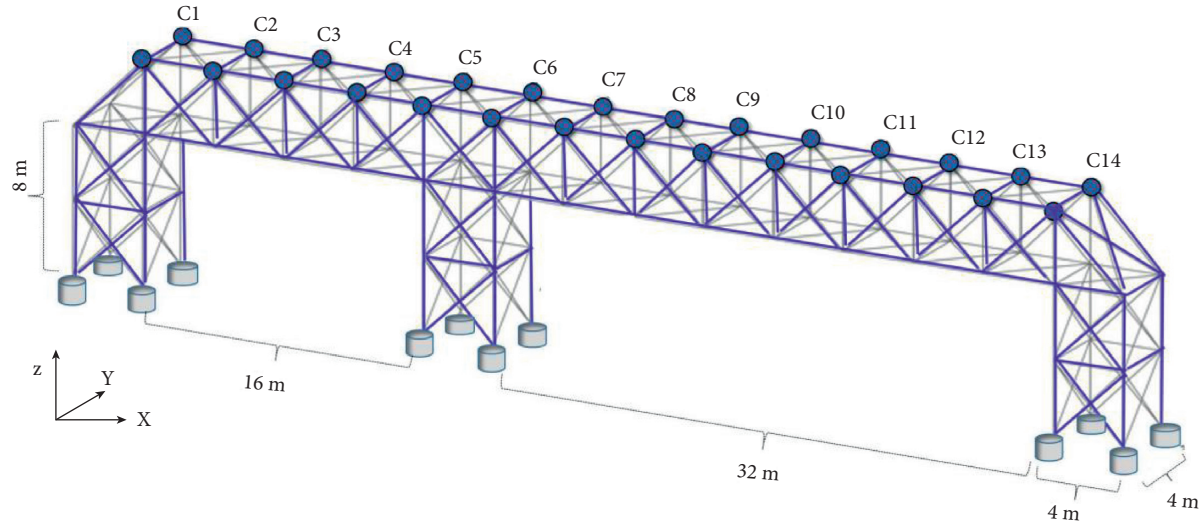

Figure 1: 3D steel truss structure with 314 bars and 84 nodes. The whole decision variables include 14 shape (C) and 314 sizing (A) elements. The shape variables show by Ci and total number of decision variables is 328.

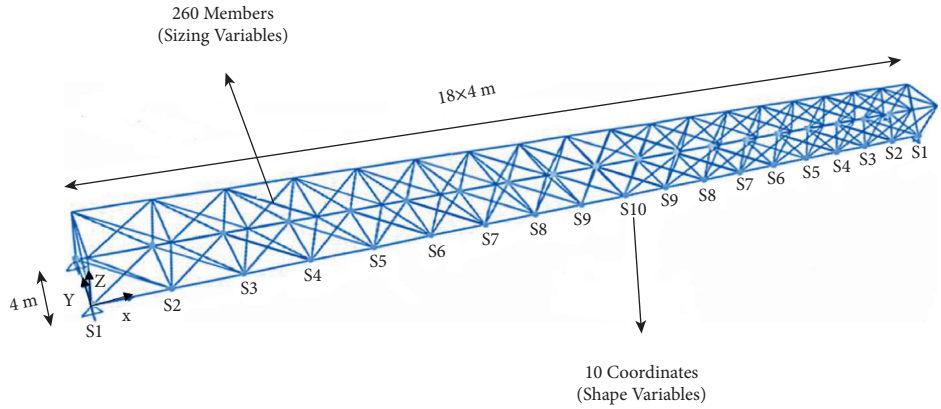

Figure 2: 3D steel truss structure with 260 bars and 76 nodes. The whole decision variables include 10 shape and 260 sizing elements. The shape variables show by Ci and total number of decision variables is 270.

two various parameters, cross-sectional and nodal coordinates which makes a heterogeneous search space. In order to evaluate the performance of the modern metaheuristics proposed in the last years, we developed a comparative platform of the twelve famous bio-inspired optimisation methods.

There are two most significant features of each bio-inspired search algorithm, including diversification and intensification. In the first step, in order to explore the search space and produce diverse solutions globally, an optimisation algorithm should be developed by a strong diversification technique. However, we need an alternative strategy to converge to a suitable solution that is the intensification process. In this step, the search focuses on the local areas to exploit and improve the current solutions. In this study, we applied and evaluated a wide range of optimisation methods with specific characteristics in order to propose the best-performed truss optimizer. Table 1 shows the initial control parameters of the optimisation methods applied in this study based on the recommended in the literature. As in this work we evaluated and compared 12 optimisation methods, we just focus on the development of best performing method in this section to avoid lengthy discussions. In order to find more technical details about the algorithms applied, we refer to Ref. [11–22].

4.1. Marine Predators Algorithm (MPA). Faramarzi et al. [19] introduced MPA, one recently evolved nature-inspired meta-heuristic algorithms, to address optimisation problems. This algorithm's basic structure focuses on mimicking various foraging patterns by ocean predators and their optimal behaviours in attempting to deal with this biological situation. For optimal foraging, marine predators typically employ two strategies: (i) Levy flight and (ii) Brownian motions. To choose between these two strategies, predators calculate the ratio of the prey's velocity to their own.

MPA's primary goal is to provide a practical and straightforward meta-heuristic algorithm based on marine predator foraging patterns. Like other (population-based)

Table 1: The configuration details of optimisation methods applied the truss shape and sizing problem. Npop is the initial population size.

```
%-----------------------------
\begin{table}(H)
\Centering
\Caption{ The configuration details of optimisation methods applied the truss shape and sizing problem. $N_{pop}$ is the initial population size.}
\label{table:meta-details}
\scalebox{0.9}{
\begin{tabular}{|l|l|l|p{6cm}|}
\hline
& \textbf{Name} & \textbf{$N_{pop}$} & \textbf{Predefined Settings} \\ \hline
1 & Grey Wolf Optimizer (GWO)~\cite{mirjalili2014grey} & 50 & $\alpha$ decreases linearly from 2 to 0 \\\hline
2 & Moth Flame Optimizer (MFO)~\cite{mirjalili2015moth} & 50 & $\alpha$ linearly dicreases from -1 to -2 \\\hline
3 & Multi Verse Optimizer (MVO)~\cite{mirjalili2016multi} & 50 & minimum and maximum of Wormhole existence probability: WEP$_{Max} = 1$,~ WEP$_{Min} = 0.2$, $\rho = 6$. \\\hline
4 & Dragonfly Algorithm (DA)~\cite{mirjalili2016dragonfly} & 50 &$w = 0.9–0.2$, $s = 0.1$, $a = 0.1$, $c = 0.7$, $f = 1$, $e = 1$. \\\hline
% 5 & Sine Cosine Algorithm (SCA)~\cite{mirjalili2016sca} & 50 & $\alpha = 2$; $r_1 = \alpha$ decreases linearly from $\alpha$ to 0 \\\hline
5 & Henry Gas Solubility optimisation (HGSO)~\cite{hashim2019henry}& 50& $N_g = 5$, $l_1 = 0.05$, $l_2 = 100$, $l_3 = 0.01$, $\alpha = 1$, $\beta = 1$, $c_1 = 0.1$, $c_2 = 0.2$ \\\hline
6 & Equilibrium Optimizer (EO)~\cite{faramarzi2020equilibrium} & 50 & $\omega_1 = 1,~\omega_2 = 2$, $GP = 0.5$(=generation probability), $V = 1$. \\\hline
7 & Arithmetic optimisation Algorithm (AOA)~\cite{abualigah2021arithmetic} & 50 & $MOP_{Max} = 1$, $MOP_{Min} = 0.2$, $C_{iter} = 1$, $\alpha = 5$, $\mu = 0.499$ \\\hline
8 & Generalized Normal Distribution (GNDO)~\cite{zhang2020generalized} & 50 & applied the default settings. \\\hline

9 & Salp Swarm Algorithm (SSA)~\cite{mirjalili2017salp} & 50& $c_1$ decreased from 2 to zero. $c_2 = rand$ and $c_3 = rand$ \\\hline

10 &Marine Predators Algorithm (MPA)~\cite{faramarzi2020marine} & 50& $p = 0.5$, $FAD = 0.2$ \\\hline
11 &Neural Network Algorithm (NNA)~\cite{sadollah2018dynamic} & 50& pre-defined settings \\\hline
12 &Water Cycle Algorithm (WCA)~\cite{eskandar2012water} & 50& $N_{sr} = 4$, $D_{max} = 10^{-5}$ \\\hline
\end{tabular}
}
\end{table}
%--------------------
```

algorithms, this algorithm begins by generating a random population in the problem searching space. As demonstrated by the survival of the fittest theory, efficient predators are always better off in terms of foraging in nature. The best solution is used in the MPA to construct an elite matrix. This matrix's arrays contain prey position information for searching and finding prey [19]. The following is the definition of the elite matrix:

```
\begin{equation}
E = \left[ {\begin{array}{cccc} {X_{1,1}^k}&{X_{1,2}^k}& \ldots &{X_{1,d}^k}
{X_{2,1}^k}&{X_{2,2}^k}& \ldots &{X_{2,d}^k}
\vdots & \vdots & \vdots & \vdots
{X_{n,1}^k} & {X_{n,2}^k} & \ldots & {X_{n,d}^k} \end{array}} \right].
\end{equation}
```

Here, $\overrightarrow{X^k}$ (top predator's vector) is replicated n times in order to build an elite matrix. Furthermore, in this equation, n represents the number of search agents, while d represents the size of the dimensions. Predators and prey are regarded as search agents in MPA. Predators and prey are regarded as search agents in MPA. In other words, when a predator looks for his prey, the prey looks for food as well. The food chain in nature-inspired this trend. The strongest predator is at the top of the food chain, and the weakest predators are subdivided into the stronger predator group. Obviously, when a stronger predator appears in this chain, this predator is moved to the top of the chain and replaces the previous hunter. The MPA imitates this concept by updating the elite matrix.

Prey is the name of another MPA matrix. This matrix has dimensions similar to the Elite matrix, and hunters adjust their positions based on it. More specifically, the initial MPA population is recognized as prey, while the best ones are selected as predators, forming the elite matrix. The prey matrix looks like this:

```
\begin{equation}
py = \left[ {\begin{array}{cccc} {{X_{1, 1}}}&{{X_{1, 2}}}& \ldots &{{X_{1, d}}}
{{X_{2, 1}}}&{{X_{2, 2}}}& \ldots &{{X_{2, d}}}
\vdots & \vdots & \vdots & \vdots
{{X_{n, 1}}} & {.{X_{n, 2}}} & \ldots & {{X_{n, d}}} \end{array}} \right].
\end{equation}
```

Here, $X_{(i,j)}$ depicts the ith prey with jth dimension. In general, the entire optimisation process in MPA is dependent on these two matrices.

Given the various phases and patterns of hunting for both marine predators and prey and the impact of the predator and prey speed on the modelling of this process, the MPA algorithm is divided into three significant steps. The following are the steps:

(i) High-velocity: during this phase, the prey's speed exceeds that of the predator

(ii) Unity-velocity: the speed of the predator and prey would be the same in this phase

(iii) Low-velocity: the prey is slower than the predator at this phase

Although predator and prey movements in nature follow unique rules and inspire the main phases of the algorithm, the MPA assigns the specified number of iterations to these phases.

4.1.1. Phase 1: (High-Velocity). This phase is used in early MPA iterations where the prey outruns the predators. Predators have the least movement during this phase, so staying in their positions is the best strategy. The MPA defines this phase as follows:

```
\begin{equation}
{\text{while}}\;\\t < \frac{1}{3}*{t_{{\rm
{max}}}}\longrightarrow.
{{\overrightarrow{Step}}_i} = {{\vec{R}}_B}\oti-
mes\left({{{\overrightarrow{Elite}}_i}-{{\vec{R}}
_B}\otimes\overrightarrow{prey
{_i}}}\right)\longrightarrow.
{\overrightarrow{prey}_i} = {\overrightarrow{prey}
_i} + P*\vec{R} \otimes {\overrightarrow{Step}_i}
\end{equation}
```

where R_B is a random number generated using the normal distribution and displaying Brownian motion, the ⊗ symbol represents entry-wise multiplications. Prey movements are simulated by multiplying R_B by prey. The uniform random numbers [0, 1] are arranged in a vector, and the constant number 0.5 is assigned to P. t and $t_{\max}$ are the current and maximum iterations, respectively. This phase takes place when the algorithm's initial iterations necessitate a high level of exploration capability.

4.1.2. Phase 2: (Unity-Velocity). Predators and prey move at the same speed in the second phase of the MPA algorithm. This stage in nature indicates that they are both looking for their own prey. The occurrence of this phase in the middle of the optimisation process demonstrates the algorithm's early stages of transition from exploration to exploitation. The MPA characterizes the prey for exploration and the predators for exploitation in this specific instance. Furthermore, during this phase, the prey moves according to the Levy theorem and the predators according to Brownian motions. This phase will be modelled in the following manner:

```
%----------
\begin{equation}
{\text{while}}\;\\frac{1}{3}*{t_{{\rm{max}}}}
< \t < \frac{2}{3}*{t_{{\rm{max}}}}.
\end{equation}
\begin{equation} \label{eq:phase2-1}
{\overrightarrow{Step}_i} = {\vec{R}_L}\otimes
\left({{{\overrightarrow{Elite}}_i}-{{\vec{R}}_L}\oti-
mes\overrightarrow{prey{_i}}} \right) \longrightarrow.
{\overrightarrow {prey} _i} = {\overrightarrow {prey}
_i} + P*\vec{R} \otimes\{\overrightarrow{Step}_i}.
\end{equation}
\begin{equation} \label{eq:phase2-2}
{\overrightarrow{Step}_i} = {\vec{R}_B}\oti-
mes\left({{{\vec{R}}_B}\otimes{{\overrightarrow
{Elite}}_i}-\overrightarrow{{prey_i}}}\right)
\longrightarrow.
{\overrightarrow {prey} _i} = {\overrightarrow{Elite}_
{{i}_i}} + P*{\rm{CF}} \otimes \ {\overrightarrow{Step}
_i}.
\end{equation}
%----------
```

It is critical to emphasize that equation (8) is applied to the first half of the MPA population. This equation $\overrightarrow{R_L}$ generates random numbers based on the Levy distribution to simulate the Levy movement [43]. In the Levy strategy, the multiplications of $\overrightarrow{R_L}$ and prey are used to simulate the movements of prey. The MPA's exploitation phase is performed using the strategy introduced in this phase. Equation (9) is how the second part of this step is simulated for the rest of the population:

In equation (10), CF is used as an adaptive parameter to control the step size of the predator's movement. Furthermore, in the Brownian strategy, the predator's movement is simulated by multiplying $\overrightarrow{R_B}$ by Elite. The position of the prey is improved as a result of the predator's Brownian movements during this phase.

```
%----------
\begin{equation}
\label{eq:cf}
{\rm{CF}} = {\left({1 - \frac{t}{{{t_{{\rm{max}}}}}}}
\right)^{\left({2\frac{t}{{{t_{{\rm{max}}}}}}} \right)}}.
\end{equation}
%----------
```

4.1.3. Phase 3: (Low-Velocity). This process was simulated in order to provide MPA with a high level of exploitation potential. This step is put in place in the algorithm when the prey's speed is slower than the predator's. Predators employ

Levy's strategy to ensure that this process is carried out correctly in the MPA. This procedure is modelled as follows:

%----------

\begin{equation}

{\text{while}}\;{\text{it}} > \frac{2}{3}*{\rm{max_{iter}}} \longrightarrow.

{{\overrightarrow{Step}}_i} = {{\vec{R}}_L}\otimes\left({{{\vec{R}}_L}\otimes{{\overrightarrow{Elite}}_i}-\overrightarrow{prey_i}}\right)\longrightarrow.

{\overrightarrow{prey}_i} = {\overrightarrow{Elite}_i} + P*{\rm{CF}}\otimes\{\overrightarrow{Step}_i}

\end{equation}

%----------

In the Levy strategy, predator movement is defined by the multiplication of $\overrightarrow{R_L}$ and Elite. Besides, Elite includes a step size to mimic predator movement in this equation and assist prey in updating their position.

4.1.4. Eddy Formation and FAD's Effect. Several factors could influence marine predators' foraging patterns in general. Environmental issues are one of the significant factors that can have a crucial impact on the behaviour of these predators. Eddy currents and fish aggregating devices (FAD) are two major environmental issues in these behavioural changes. Filmalter et al. [43] discovered that sharks spend the vast majority of their hunting time (nearly 80%) near FADs in the wild. Furthermore, the remainder of the shark hunting time is spent locating areas with specific prey distributions. These FADs are regarded as local optimum points in the MPA, capable of trapping the algorithm. As a result, the effect of FADs on the MPA algorithm is as follows:

\begin{equation}

{\overrightarrow {prey} _{\rm{i}}} = \\

\left\{ {\begin{array}{ll} {{\overrightarrow {prey} }_i} + {\rm{CF}}\left[{{{\vec{X}}_{min}} + R \otimes ({{\vec{X}}_{max}} - \ {{\vec{X}}_{min}}} \right)] \otimes \vec{U} & {\text{if}}\ r < {\rm{FADs}}\\

{{\overrightarrow {prey} }_i} + \left[{{\rm{FADs}}\left({1 - r} \right) + r} \right]\left({{{\overrightarrow {prey} }_{r_1}} - {{\overrightarrow {prey} }_{r_2}}} \right) & {\text{if}}\ r > {\rm{FADs}} \end{array}} \right.

\end{equation}

%----------

Here, $\overrightarrow{X}_{\min}$ and $\overrightarrow{X}_{\max}$ are the lower and upper bounds of the dimensions, respectively. $\overrightarrow{U}$ creates a binary vector consisting of zero and one array. The FAD factor is the probability of FADs influencing the optimisation process, and its value is set to 0.2. On the other hand, r is defined as a uniform random number that generates values between [0, 1]. Subscription of r1 and r2 denotes random prey matrix indexes.

5. Experimental Results

In this section, we demonstrate the optimisation outcomes achieved by twelve state-of-the-art meta-heuristics in order to minimize the total weight of two truss structures. For all search algorithms, the originally recommended parameters with the same population size were applied to provide a fair comparison framework.

In Figure 3, each curve represents the development of the average truss's weight plus the penalty of 314-bar case study that yielded by the best-found design for each optimisation method over 105 evaluation iterations. From this convergence plot, we can observe that the optimisation methods can be classified into two groups. First, optimisation strategies with a high convergence rate. These methods have a heightened exploitation ability include AOA, DA, GNDO, MPA, HGSO, and SSA. In the second group, we can see four methods with slow convergence speed, such as GWO, EO, MFO and MVO. In this case study (314-bar), the fastest convergence rate is related to the MPA, which could find a relative-optimal structure; however, it struggled with a local optimum and could not escape from this situation. To have a general observation from Figure 3, except for DA, all methods in the second group were converged to a local optimum design in the same iteration approximately. The DA could improve the best-found solutions after 4×10^4 evaluation number.

In order to provide an accurate comparison framework for these twelve optimisation methods' performance, we plotted Figure 4. In this figure, each box shows the 25% and 75% percentiles of the best-found solutions (upper and lower edges) and the median of the outcomes represented by the central tag. On each box, we can see the extended whiskers for the data points with larger variance, and finally, the outliers are depicted by the '+' symbol. From Figure 4, the most important observation is that both MPA and DA are the best-performed optimisation methods compared with other algorithms.

A comparison of the convergence rate of twelve optimizers for the 260-bar truss can be seen in Figure 5. The Figure depicts that SSA and GNDO have the most considerable convergence speed; however, the mean weight of 260-bar proposed by MPA can be significantly better. Interestingly, the whole methods applied for this large truss could not improve the quality of the solutions after 24 iterations. This premature convergence shows these meta-heuristics are not able to exit from the local optima. The highly dynamic constraints and large number of decision variables of the this truss problem can be the most significant reason for the premature convergence issue. The statistical results of the best-found solutions for twelve algorithms summarize in Figure 6. The Figure illustrates that both MPA and AOA could propose the best configuration of 260-bar compared with others methods applied. Moreover, the performance of the SSA and GNDO is considerable. In the 260-bar case study, the DA's exploration and exploitation strategies were not effective in figuring out the challenging constraints of the problem.

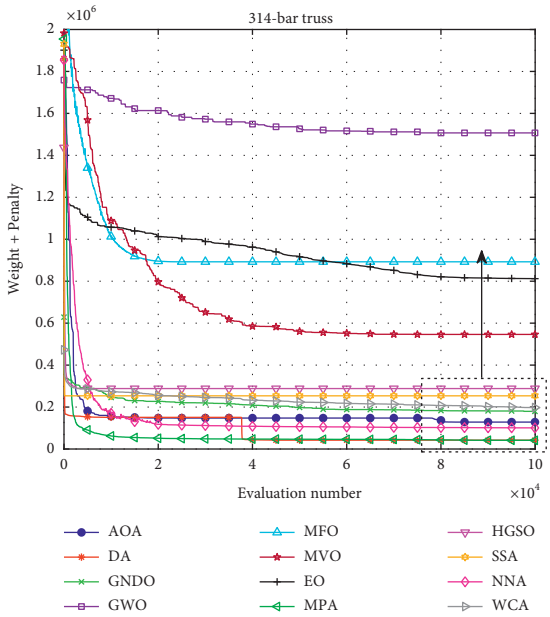

Figure 3: A convergence rate comparison of twelve optimisation methods applied for the large-scale 314-bar truss problem. The maximum number of evaluation is 105. Total number of decision variables is 328.

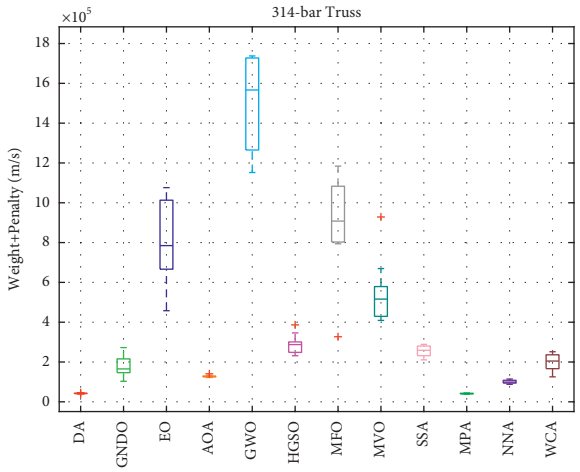

Figure 4: box plot of twelve various optimisation methods' performance for 314-bar truss problem.

The statistical performance (best, worst, average, median and standard deviation) of these optimisation methods is quantified in Table 2 for both case studies, 314-bar and 260-bar. It can be seen that the best-found designs were proposed by the MPA in both truss problems. It is noted that the reported solutions in Table 3 are an accumulated structure weight and the penalty value. The second best method is the AOA on average in both truss problems.

In order to tune a proper population size for the MPA, we tested four population sizes consisting of 50, 100, 250,

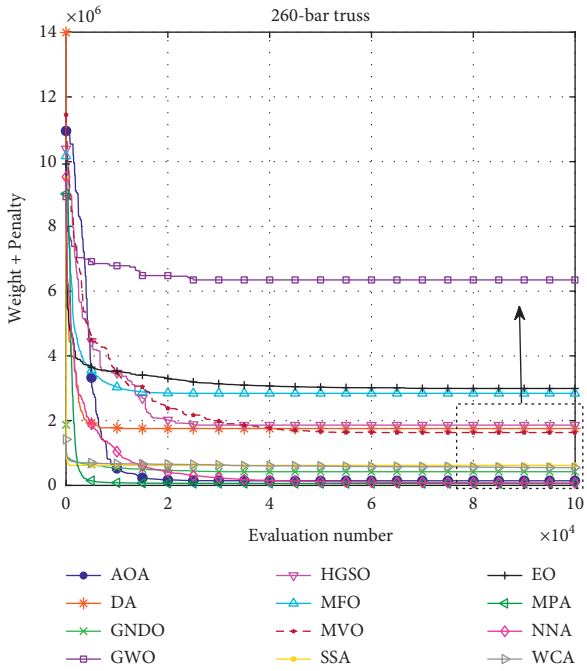

Figure 5: A convergence rate comparison of twelve optimisation methods applied for the large-scale 260-bar truss problem. The maximum number of evaluation is 105. Total number of decision variables is 270.

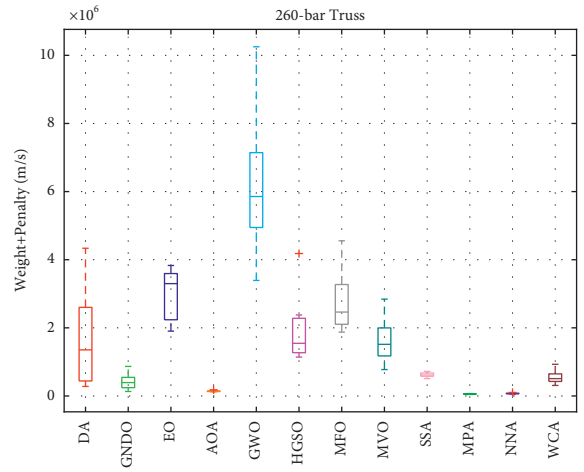

Figure 6: Statistical analysis of the optimisation methods' performance for 260-bar truss problem. The best-found solution per each run.

and 500 with the same evaluation number at 105. Figure 7 shows the convergence speed of this competition among four sizes of the MPA population. It can be seen that the performance of MPA with a population size of 100 outweighs other settings in terms of designs weight. The significant observation from Figure 7 is that the population size of 50 cannot be an efficient setting for the MPA and wakened the optimisation process.

Table 2: The statistical results of the best-found truss structures for 12 optimisation methods.

```
\begin{table}[]
\centering
\caption{ The statistical results of the best-found truss structures for 12 optimisation methods.}
\label{table:best_found}
\scalebox{0.7}{
\begin{tabular}{lllllllllllll}
\hlineB{4}
& \multicolumn{10}{c}{260-bar truss} & \multicolumn{1}{c}{} & \multicolumn{1}{c}{} \\ \hlineB{3}
& DA & GNDO & EO & AOA & GWO & HGSO & MFO & MVO & SSA & MPA & NNA & WCA \\ \hlineB{4}
Min & 2.793E+05 & 1.318E+05 & 1.904E+06 & 1.204E+05 & 3.392E+06 & 1.141E+06 & 1.874E+06 & 7.753E+05 & 5.091E+05 & \textbf{4.452E+04} & 5.537E+04 & 3.086E+05
Max & 4.336E+06 & 8.669E+05 & 3.831E+06 & 1.801E+05 & 1.025E+07 & 4.181E+06 & 4.555E+06 & 2.843E+06 & 7.212E+05 & 7.031E+04 & 1.025E+05 & 9.297E+05
Mean & 1.756E+06 & 4.224E+05 & 2.994E+06 & 1.423E+05 & 6.347E+06 & 1.862E+06 & 2.844E+06 & 1.632E+06 & 6.182E+05 & 5.777E+04 & 7.305E+04 & 5.485E+05
Median & 1.352E+06 & 3.930E+05 & 3.298E+06 & 1.399E+05 & 5.855E+06 & 1.545E+06 & 2.464E+06 & 1.516E+06 & 6.146E+05 & 5.836E+04 & 6.973E+04 & 5.091E+05
STD & 1.482E+06 & 2.372E+05 & 7.436E+05 & 1.811E+04 & 2.029E+06 & 9.217E+05 & 1.002E+06 & 6.528E+05 & 6.302E+04 & 7.729E+03 & 1.306E+04 & 1.781E+05 \\ \hlineB{4}
& \multicolumn{10}{c}{314-bar truss} & \multicolumn{1}{c}{} & \multicolumn{1}{c}{} \\ \hlineB{3}
& DA & GNDO & EO & AOA & GWO & HGSO & MFO & MVO & SSA & MPA & NNA & WCA \\\hlineB{3}
Min & 3.934E+04 & 1.031E+05 & 4.578E+05 & 1.223E+05 & 1.152E+06 & 2.315E+05 & 3.269E+05 & 4.089E+05 & 2.107E+05 & \textbf{3.888E+04} & 8.769E+04 & 1.254E+05
Max & 4.598E+04 & 2.723E+05 & 1.076E+06 & 1.408E+05 & 1.738E+06 & 3.864E+05 & 1.184E+06 & 9.286E+05 & 2.885E+05 & 4.430E+04 & 1.145E+05 & 2.515E+05
Mean & 4.217E+04 & 1.795E+05 & 8.122E+05 & 1.283E+05 & 1.507E+06 & 2.882E+05 & 8.925E+05 & 5.463E+05 & 2.536E+05 & 4.145E+04 & 1.006E+05 & 1.975E+05
Median & 4.192E+04 & 1.653E+05 & 7.848E+05 & 1.279E+05 & 1.567E+06 & 2.875E+05 & 9.080E+05 & 5.161E+05 & 2.590E+05 & 4.121E+04 & 9.906E+04 & 2.048E+05
STD & 1.647E+03 & 5.375E+04 & 2.141E+05 & 5.298E+03 & 2.303E+05 & 4.896E+04 & 2.389E+05 & 1.602E+05 & 2.784E+04 & 1.697E+03 & 8.896E+03 & 4.533E+04 \\ \hlineB{4}
\end{tabular}
}
\end{table}
%---------------
```

Table 3: Shows the average sum-rank Friedman test for the 314-bar and 260-bar problems using twelve optimisation methods. It can be observed that the MPA achieved the first rank in both case studies on average, and in the following the NNA, AOA and GNDO placed the second, third, and fourth rank of the truss optimisation.

```
\begin{table}[]
\centering
\caption{ The average performance rank of 12 optimisation methods based on the best-found designs. significant level of the Friedman test is 0.05.}
\label{table:rank}
\scalebox{0.8}{
\begin{tabular}{l|l|l|l|l|l|l|l|l|l|l|l|l}
\hlineB{4}
& {DA} & {GNDO} & {EO} & {AOA} & {GWO} & {HGSO} & {MFO} & {MVO} & {SSA} & {MPA}&{NNA} & {WCA}\\ \hlineB{3}
314-bar & 2&5&10&4&12&8&11&9&7&1&3&6 \\ \hline
260-bar & 8&4&11&3&12&9&10&7&6&1&2&5 \\ \hline
Mean rank & 5&4.5&10.5&3.5&12&8.5&10.5&8&6.5&1&2.5&5.5 \\\hlineB{4}
\end{tabular}
}
\end{table}
```

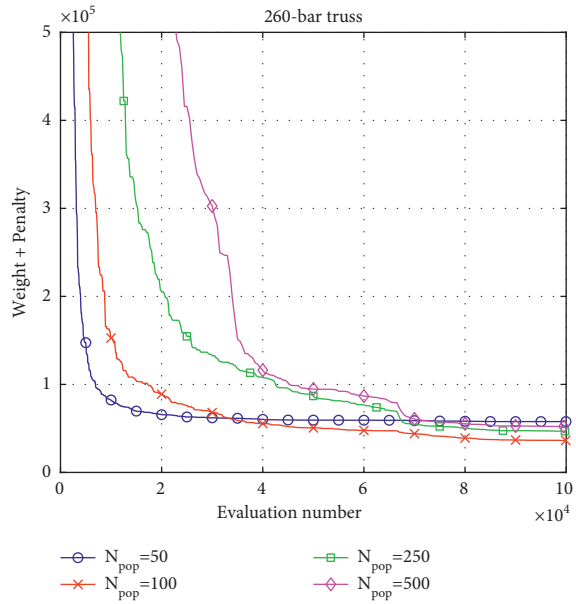

Figure 7: A convergence rate comparison of four different population sizes of MPA optimisation method applied for the large-scale 260-bar truss problem. The maximum number of evaluation is 105. A zoom version of the figure can be seen in top right side.

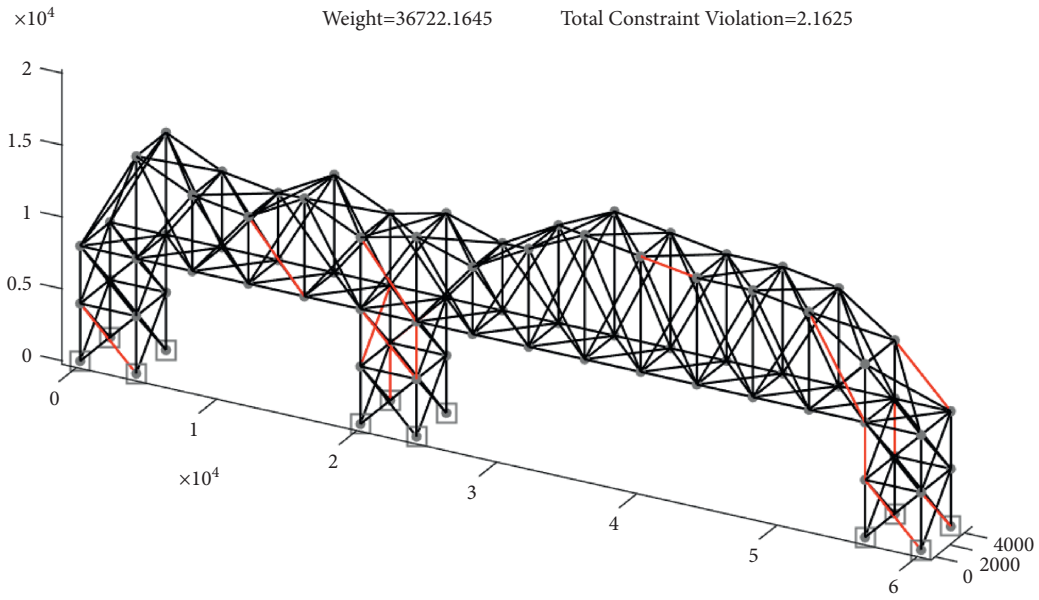

Figure 8: Best-found design of 314-bar structure. The violated bars are highlighted by red color.

In order to show the best-found structure of two case studies, 314, and 260-bar trusses, Figure 8 and Figure 9 are plotted. In both designs, we can see a few number of bars that are violated under stress and displacement forces (highlighted by red colour). The total sum violations are low in both cases; however, in the real applications, these violations should be minimised as much as possible near to zero. This comparative optimisation framework obviously shows that the meta-heuristics applied in this study need to be improved in terms of the constraint handling methods specifically.

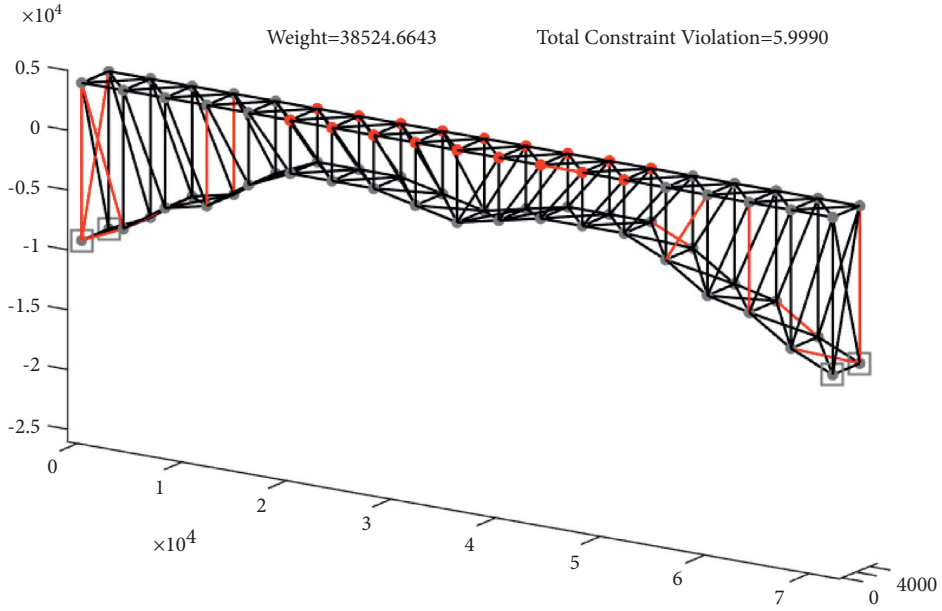

Figure 9: Best-found design of 260-bar structure. The violated bars are highlighted by red color.

6. Conclusion

This paper used 12 modern meta-heuristic algorithms to consider the truss shape and sizing optimisation problem. To handle the violation of constraints, we applied a penalty function that is a popular method in this way. Different penalty factors were evaluated to find the best value in terms of best-found designs. Two various truss problems are used in this study. Both of them have a large structure composed of 314 and 260 bars, respectively. It is assumed that the topology of the truss should be fixed and unalterable. The optimal truss shape and sizing variables should be obtained by minimizing the structural weight with respect to nodal displacement constraints, element stress constraints, and natural frequencies. These complex constraints make a challenging optimisation problem, which is large-scale, nonlinear, multimodal with dynamic constraints. As global optimisation algorithms are mostly efficient and robust, we mainly focus on the application of metaheuristics, especially modern swarm optimisation methods, to the truss optimisation problems in this study. Indeed, we applied twelve different bio-inspired optimisation methods in order to evaluate and develop a comparative framework for the large-scale truss problems. To have a fair comparison, all control parameters were set according to literature recommendations for each optimisation algorithm.This is because there is no simple way to obtain the best parameter values. From an engineering point of view, the performance of the MPA method is better than other optimisation methods used in this study because of a combination of fast and effective exploration and exploitation search strategies. Furthermore, we tuned the population size of the MPA and showed that 100 could be a better option than other dimensions.

Data Availability

In this study, we did not apply a specific dataset.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] L. Wei, T. Tang, X. Xie, and W. Shen, "Truss optimization on shape and sizing with frequency constraints based on parallel genetic algorithm," *Structural and Multidisciplinary Optimization*, vol. 43, no. 5, pp. 665–682, 2011.

[2] B. Doerr and F. Neumann, "A survey on recent progress in the theory of evolutionary algorithms for discrete optimization," *ACM Transactions on Evolutionary Learning and Optimization*, vol. 1, no. 4, pp. 1–43, 2021.

[3] M. Khatibinia, S. Sadegh Naseralavi, and S. Sadegh Naseralavi, "Truss optimization on shape and sizing with frequency constraints based on orthogonal multi-gravitational search algorithm," *Journal of Sound and Vibration*, vol. 333, no. 24, pp. 6349–6369, 2014.

[4] C. Renkavieski and R. Stubs Parpinelli, "Meta-heuristic algorithms to truss optimization: literature mapping and application," *Expert Systems with Applications*, vol. 182, Article ID 115197, 2021.

[5] J. Kennedy and E. Russell, "Particle swarm optimization," *Proceedings of ICNN'95-international conference on neural networks*, IEEE, vol. 4, pp. 1942–1948, 1995.

[6] A. Kaveh and A. Zolghadr, "Democratic pso for truss layout and size optimization with frequency constraints," *Computers & Structures*, vol. 130, pp. 10–21, 2014.

[7] J. A. Bland, "Optimal structural design by ant colony optimization," *Engineering Optimization*, vol. 33, no. 4, pp. 425–443, 2001.

[8] Q. X. Lieu, D. Tt Do, and J. Lee, "An adaptive hybrid evolutionary firefly algorithm for shape and sizeoptimization of truss structures with frequency constraints," *Computers & Structures*, vol. 195, pp. 99–112, 2018.

[9] M. Arjmand, M. Sheikhi Azqandi, and M. Delavar, "Hybrid improved dolphin echolocation and ant colony optimization for optimal discrete sizing of truss structures," *Journal of rehabilitation in civil engineering*, vol. 6, no. 1, pp. 70–87, 2018.

[10] A. Kaveh and P. Zakian, "Improved gwo algorithm for optimal design of truss structures," *Engineering with Computers*, vol. 34, no. 4, pp. 685–707, 2018.

[11] S. Mirjalili, S. Mohammad Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[12] S. Mirjalili, "Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm," *Knowledge-Based Systems*, vol. 89, pp. 228–249, 2015.

[13] S. M. Mirjalili and A. Hatamlou, "Multi-Verse Optimizer: a nature-inspired algorithm for global optimization," *Neural Computing & Applications*, vol. 27, no. 2, pp. 495–513, 2016.

[14] S. Mirjalili, "Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural Computing & Applications*, vol. 27, no. 4, pp. 1053–1073, 2016.

[15] A. Faramarzi, M. Heidarinejad, B. Stephens, and S. Mirjalili, "Equilibrium optimizer: a novel optimization algorithm," *Knowledge-Based Systems*, vol. 191, Article ID 105190, 2020.

[16] L. Abualigah, A. Diabat, S. M. Mohamed, A. Elaziz, and A. H. Gandomi, "The arithmetic optimization algorithm," *Computer Methods in Applied Mechanics and Engineering*, vol. 376, Article ID 113609, 2021.

[17] Y. Zhang, Z. Jin, and S. Mirjalili, "Generalized normal distribution optimization and its applications in parameter extraction of photovoltaic models," *Energy Conversion and Management*, vol. 224, Article ID 113301, 2020.

[18] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp swarm algorithm: a bio-inspired optimizer for engineering design problems," *Advances in Engineering Software*, vol. 114, pp. 163–191, 2017.

[19] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, "Marine predators algorithm: a nature-inspired metaheuristic," *Expert Systems with Applications*, vol. 152, Article ID 113377, 2020.

[20] F. A Hashim, E. H Houssein, M. S Mabrouk, W. Al-Atabany, and S. Mirjalili, "Henry gas solubility optimization: a novel physics-based algorithm," *Future Generation Computer Systems*, vol. 101, pp. 646–667, 2019.

[21] A. Sadollah, H. Sayyaadi, and A. Yadav, "A dynamic metaheuristic optimization model inspired by biological nervous systems: Neural network algorithm," *Applied Soft Computing*, vol. 71, pp. 747–782, 2018.

[22] H. Eskandar, A. Sadollah, A. Bahreininejad, and M. Hamdi, "Water cycle algorithm - a novel metaheuristic optimization method for solving constrained engineering optimization problems," *Computers & Structures*, vol. 110-111, pp. 151–166, 2012.

[23] H. Rahami, A. Kaveh, and Y. Gholipour, "Sizing, geometry and topology optimization of trusses via force method and genetic algorithm," *Engineering Structures*, vol. 30, no. 9, pp. 2360–2369, 2008.

[24] V. Ho-Huu, T. Nguyen-Thoi, T. Khac, L. Anh, and T. Vo-Duy, "An improved differential evolution based on roulette wheel selection for shape and size optimization of truss structures with frequency constraints," *Neural Computing and Applications*, vol. 29, no. 1, pp. 167–185, 2018.

[25] Sy. Nguyen-Van, K. T. Nguyen, V. Hai Luong, S. Lee, and Q. X. Lieu, "A novel hybrid differential evolution and symbiotic organisms search algorithm for size and shape optimization of truss structures under multiple frequency constraints," *Expert Systems with Applications*, vol. 184, Article ID 115534, 2021.

[26] L. Lamberti, "An efficient simulated annealing algorithm for design optimization of truss structures," *Computers & Structures*, vol. 86, no. 19-20, pp. 1936–1953, 2008.

[27] S. Kazemzadeh Azad, "Enhanced hybrid metaheuristic algorithms for optimal sizing of steel truss structures with numerous discrete variables," *Structural and Multidisciplinary Optimization*, vol. 55, no. 6, pp. 2159–2180, 2017.

[28] S. Ali, A. Bahreininejad, H. Eskandar, and M. Hamdi, "Mine blast algorithm for optimization of truss structures with discrete variables," *Computers & Structures*, vol. 102, pp. 49–63, 2012.

[29] F. K. J. Jawad, C. Ozturk, W. Dansheng, M. Mahmood, O. Al-Azzawi, and A. Al-Jemely, "Sizing and layout optimization of truss structures with artificial bee colony algorithm," *Structures*, Elsevier, vol. 30, pp. 546–559, 2021.

[30] M. Sonmez, "Artificial bee colony algorithm for optimization of truss structures," *Applied Soft Computing*, vol. 11, no. 2, pp. 2406–2418, 2011.

[31] A. Kaveh, S. Talatahari, and N. Khodadadi, "Hybrid invasive weed optimization-shuffled frog-leaping algorithm for optimal design of truss structures," *Iranian Journal of Science and Technology, Transactions of Civil Engineering*, vol. 44, 2019.

[32] A. Kaveh and M. Khayatazad, "A new meta-heuristic method: ray optimization," *Computers & Structures*, vol. 112-113, pp. 283–294, 2012.

[33] A. Kaveh and M. I. Ghazaan, "Layout and size optimization of trusses with natural frequency constraints using improved ray optimization algorithm," *Iranian Journal of Science and Technology Transactions of Civil Engineering*, vol. 39, no. 2, pp. 395–408, 2015.

[34] S. Talatahari, A. H. Gandomi, and G. J. Yun, "Optimum design of tower structures using firefly algorithm," *The Structural Design of Tall and Special Buildings*, vol. 23, no. 5, pp. 350–361, 2014.

[35] K. Ali and N. Farhoudi, "A new optimization method: dolphin echolocation," *Advances in Engineering Software*, vol. 59, pp. 53–70, 2013.

[36] T. Dede and Y. Ayvaz, "Combined size and shape optimization of structures with a new meta-heuristic algorithm," *Applied Soft Computing*, vol. 28, pp. 250–258, 2015.

[37] R. Awad, "Sizing optimization of truss structures using the political optimizer (po) algorithm," *Structures*, Elsevier, vol. 33, pp. 4871–4894, 2021.

[38] A. Kaveh and S. Talatahari, "Optimum design of skeletal structures using imperialist competitive algorithm," *Computers & Structures*, vol. 88, no. 21-22, pp. 1220–1229, 2010.

[39] M. Dehghani, M. Mashayekhi, and M. Sharifi, "An efficient imperialist competitive algorithm with likelihood assimilation for topology, shape and sizing optimization of truss structures," *Applied Mathematical Modelling*, vol. 93, no. 1–27, 2021.

[40] C. A. C. Coello, "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art," *Computer Methods in Applied Mechanics and Engineering*, vol. 191, no. 11-12, pp. 1245–1287, 2002.

[41] S. Kazemzadeh Azad and O. Hasançebi, "Structural optimization problems of the iscso 2011-2015: a test set,"

International Journal of Optimization in Civil Engineering, vol. 6, no. 4, pp. 629–638, 2016.

[42] B. Optimizer, "International Student Competition in Structural Optimization," pp. 12–25, 2021, https://www. bright-optimizer.com/.

[43] J. D. Filmalter, L. Dagorn, P. D. Cowley, and M. Taquet, "First descriptions of the behavior of silky sharks, carcharhinus falciformis, around drifting fish aggregating devices in the indian ocean," *Bulletin of Marine Science*, vol. 87, no. 3, pp. 325–337, 2011.