INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING
Int. J. Optim. Civil Eng., 2021; 11(4):663-693

IMPROVED ARITHMETIC OPTIMIZATION ALGORITHM FOR STRUCTURAL OPTIMIZATION WITH FREQUENCY CONSTRAINTS

A. Kaveh[*, †] , K. Biabani Hamedani and M. Kamalinejad

School of Civil Engineering, Iran University of Science and Technology, Tehran, Iran

ABSTRACT

The arithmetic optimization algorithm (AOA) is a recently developed metaheuristic optimization algorithm that simulates the distribution characteristics of the four basic arithmetic operations (i.e., addition, subtraction, multiplication, and division) and has been successfully applied to solve some optimization problems. However, the AOA suffers from poor exploration and prematurely converges to non-optimal solutions, especially when dealing with multi-dimensional optimization problems. More recently, in order to overcome the shortcomings of the original AOA, an improved version of AOA, named IAOA, has been proposed and successfully applied to discrete structural optimization problems. Compared to the original AOA, two major improvements have been made in IAOA: (1) The original formulation of the AOA is modified to enhance the exploration and exploitation capabilities; (2) The IAOA requires fewer algorithm-specific parameters compared with the original AOA, which makes it easy to be implemented. In this paper, IAOA is applied to the optimal design of large-scale dome-like truss structures with multiple frequency constraints. To the best of our knowledge, this is the first time that IAOA is applied to structural optimization problems with frequency constraints. Three benchmark dome-shaped truss optimization problems with frequency constraints are investigated to demonstrate the efficiency and robustness of the IAOA. Experimental results indicate that IAOA significantly outperforms the original AOA and achieves results comparable or superior to other state-of-the-art algorithms.

Keywords: metaheuristics; arithmetic optimization algorithm (AOA); improved arithmetic optimization algorithm (IAOA); structural optimization; frequency constraints; truss structures.

Received: 4 October 2021; Accepted: 14 November 2021

[*] Corresponding author: Centre of Excellence for Fundamental Studies in Structural Engineering, Iran University of Science and Technology, Narmak, Tehran, P.O. Box 16846-13114, Iran
[†]E-mail address: alikaveh@iust.ac.ir (A. Kaveh)

1. INTRODUCTION

In free vibration of a structure, natural frequencies are the fundamental characteristics affecting the dynamic behavior of the structure [1]. As an example, the dynamic response of a low frequency vibration system is mainly a function of the fundamental natural frequency of the system [2]. In such cases, the dynamic characteristics of the structure can be significantly improved by manipulating the selected frequency. This aim can be achieved through the optimal design of structures with frequency constraints. As a result, structural optimization with frequency constraints makes it possible to manipulate the dynamic characteristics in a variety of ways. For example, in designing a spaceship, it is necessary to impose constraints on several of the lowest natural frequencies of the vehicle to prescribed ranges natural in order to avoid resonance phenomenon.

Optimum design of structures considering frequency constraints has attracted the attention of many researchers since the 1980s. Bellagamba and Yang [3] applied a nonlinear programming procedure to the minimum mass design of truss structures under static thermal and mechanical loads considering various constraints, including fundamental natural frequency and local buckling. Grandhi and Venkayya [2] employed an optimality criterion method based on uniform Lagrangian density for the minimum weight design of structures with multiple frequency constraints. Tong and Liu [4] proposed a dynamically constrained optimization procedure for the optimal design of truss structures with discrete design variables under dynamic constraints. Sedaghati et al. [5] used the integrated force method as an analyzer to optimize both truss and beam structures under frequency constraints. Lingyun et al. [6] proposed a niche hybrid genetic algorithm (NHGA) for solving shape and size optimization of truss structures with multiple frequency constraints. Gomes [7] employed a particle swarm optimization (PSO) algorithm for size and shape optimization of truss structures taking into account frequency constraints. Kaveh and Zolghadr [8] proposed a hybridization of the charged system search (CSS) and the big bang-big crunch (BB-BC) algorithms with trap recognition capability and applied it to the optimization of truss structures with frequency constraints. Khatibnia and Naseralavi [1] introduced the orthogonal multi-gravitational search algorithm (OMGSA) to solve size and shape truss optimization problems with frequency constraints. Kaveh and Ilchi Ghazaan [9] performed the optimal design of large-scale dome structures with multiple natural frequency constraints. Ho-Huu et al. [10] proposed a novel differential evolution (DE) for size and shape optimization of truss structures with frequency constraints. Kaveh and Zolghadr [11] utilized the cyclical parthenogenesis algorithm (CPA) for optimal design of cyclically symmetric trusses with frequency constraints. Lieu et al. [12] proposed a hybridization of the differential evolution (DE) algorithm and the firefly algorithm (FA) for shape and size optimization of truss structures under multiple frequency constraints.

Frequency-constrained optimization problems are well-known as highly nonlinear, non-convex, and multimodal optimization problems with respect to the design variables [13]. The most common difficulty with the frequency-constrained optimization problems is the switching of vibration modes due to structural size and shape modifications, which may cause convergence difficulties [14]. As a result, classical methods of optimization based on gradients may not be effective to solve this type of optimization problems because these methods require the gradient information of the frequency with respect to the design

variables [6]. Thus, metaheuristic optimization algorithms can be regarded as appropriate alternatives.

Metaheuristic optimization algorithms represent a branch of approximate optimization techniques that have been one of the most active fields of research in computer science over the last years [15]. These techniques have the capability to find optimal or near-optimal solutions to tough optimization problems and even NP-hard problems within a reasonable computational time [16]. Metaheuristics have found a wide range of engineering applications because they: (1) are easy to design and implement; (2) use no gradient information during the optimization process; and (3) are applicable to a large variety of optimization problems.

Over the past three decades, numerous metaheuristics have been developed, most of which are nature-inspired. Nature-inspired metaheuristic algorithms can be categorized into four main groups (see Fig. 1): evolution-based, physics-based, swarm-based, and human-based [17]. Evolution-based algorithms imitate the concepts of biological evolution. Some of the most popular evolution-based algorithms are genetic algorithm (GA) [18], evolution strategy (ES) [19], genetic programming (GP) [20], evolutionary programming (EP) [21], memetic algorithm (MA) [22], and differential evolution (DE) [23]. Physics-based algorithms are inspired by the physical laws of nature. Some of the most well-known physics-based algorithms are gravitational search algorithm (GSA) [24], charged system search (CSS) [25], big bang-big crunch (BB-BC) [26], ray optimization (RO) [27], and water evaporation optimization (WEO) [28]. Swarm-based algorithms form another group of nature-inspired algorithms that are based on the social behavior of groups of animals. Some of the most popular swarm-based metaheuristic algorithms are artificial bee colony (ABC) [29], particle swarm optimization (PSO) [30], ant colony optimization (ACO) [31], and cuckoo search (CS) [32]. The last group of nature-inspired algorithms is human-based metaheuristics that mimic the human behaviors and characteristics. Harmony Search (HS) mimics the improvisation of music players [33]. League championship algorithm (LCA) simulates the competition of sport teams in a sport league [34]. Teaching-learning-based optimization (TLBO) algorithm is developed based on the philosophy of the teaching-learning process [35]. Imperialist competitive algorithm (ICA) is established inspired by the policy of extending a country's beyond its own boundaries [36]. Dynastic optimization algorithm (DOA) is developed by using social behavior in human dynasties [37]. Interior search algorithm (ISA) is established inspired by interior design and decoration [38].

The arithmetic optimization algorithm (AOA) is a newly proposed metaheuristic algorithm inspired by the four basic arithmetic operations of subtraction, addition, multiplication, and division [39]. Up to now, the AOA has been employed for solving some real-world optimization problems, including structural damage detection [40], maximum power point tracking [41], model identification [42], image segmentation [43], etc. However, it has been found that the original AOA suffers from poor exploration and is prone to trap into local optima [42-43]. Recently, in order to overcome the drawbacks of the original AOA, an improved variant of the original AOA, called improved AOA (IAOA), has been proposed and successfully applied to discrete structural optimization problems [44]. The main contributions of IAOA, as compared with the original AOA, are as follows [44]: (1) Both the exploration and exploitation capabilities of the original AOA are enhanced to overcome its drawbacks. (2) The IAOA requires less algorithm-specific parameters

compared to the original AOA, which makes it easy to be applied [44]. In this research, IAOA is applied to solve structural optimization problems with multiple frequency constraints. To our knowledge, both the original AOA are IAOA are still not applied to structural optimization with frequency constraints. The performance of the IAOA is demonstrated through three benchmark examples of frequency-constrained structural optimization problems. The results achieved by the AOA and IAOA are compared with each other and with those of some other state-of-the-art optimization methods existing in the literature.

The rest of this paper is organized as follows: In Section 2.1, the mathematical formulation of the size optimization problem of truss structures with multiple frequency constraints is presented. Section 2.2 provides a detailed discussion of the original AOA. In Section 2.3, the improved arithmetic optimization algorithm (IAOA) is reviewed in detail. In Section 3, three large-scale numerical examples are investigated and the optimization results are discussed. Finally, Section 4 draws the concluding remarks.

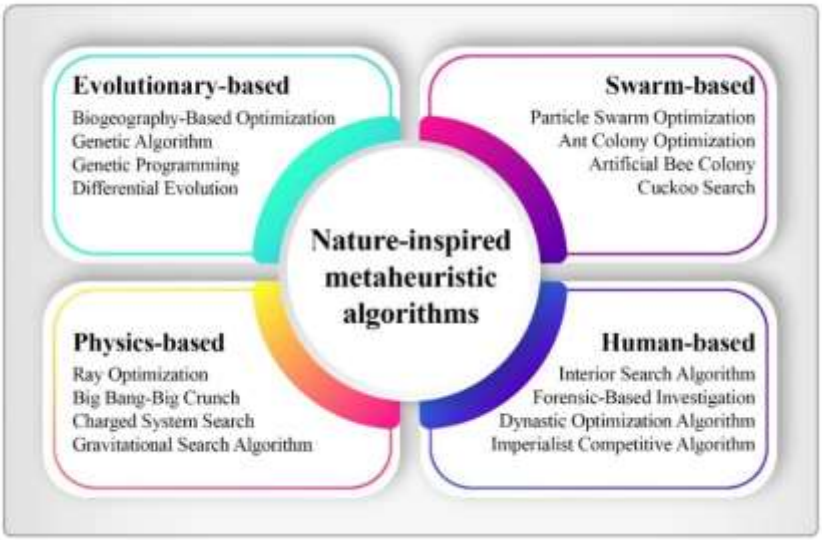

Figure 1. Classification of nature-inspired metaheuristic algorithms [44]

2. MATERIALS AND METHODS

2.1 Problem statement

In the case of truss sizing optimization with multiple frequency constraints, the objective is to minimize the weight of the structure while satisfying multiple constraints on natural frequencies [45]. For this purpose, the cross-sectional areas of members are considered as design variables. On the other hand, the layout of the structure is not changed during the design process. The problem can therefore be mathematically formulated as follows [46-47]:

$$\text{Find: } \boldsymbol{A} = \{A_1, A_2, \cdots, A_d\} \tag{1}$$

$$\text{to minimize: } f(\boldsymbol{A}) = \sum_{i=1}^{m} \rho_i A_i L_i \tag{2}$$

$$\text{subject to: } \begin{cases} \omega_j \geq \omega_j^*, & \text{for some natural frequencies } j \\ \omega_k \leq \omega_k^*, & \text{for some natural frequencies } k \\ A_{i,min} \leq A_i \leq A_{i,max}, & i = 1,2,\dots,d \end{cases} \tag{3}$$

where $\boldsymbol{A}$ represents the vector of design variables (i.e., the cross-sectional areas of members); d is the total number of design variables; m is the total number of truss members; ρ_i, A_i, and L_i are the material density, cross-sectional area, and length of the i-th truss member; $f(\boldsymbol{A})$ denotes the objective function (i.e., the total weight of the truss structure); ω_j and ω_k are the j-th and k-th natural frequencies of the truss structure, respectively; ω_j^* and ω_k^* stand for the lower and upper bounds corresponding to ω_j and ω_k, respectively; and $A_{i,min}$ and $A_{i,max}$ denote the lower and upper bounds of A_i, respectively.

The above problem is a constrained optimization problem. To deal effectively with the frequency constraints of the problem, the penalty function method, the most common constraint-handling approach, is employed. In the penalty function method, the original constrained optimization problem is transformed into an unconstrained one, by incorporating a penalty term in the objective function [15]. In this paper, the following dynamic multiplicative penalty function is adopted [48-49]:

$$f_{penalty}(\boldsymbol{A}) = (1 + \varepsilon_1 \times c)^{\varepsilon_2}, \qquad c = \sum_{i=1}^{q} c_i \tag{4}$$

where $f_{penalty}(\boldsymbol{A})$ is the penalty function; c stands for the sum of constraint violations; q is the total number of frequency constraints; and ε_1 and ε_2 are the parameters of the penalty function. These parameters are adjusted based on the exploration and exploitation properties of the search process [48]. In this study, in all design examples, ε_1 is set to a constant value, and ε_2 increases linearly with the number of iterations. This implies that, as the search progresses, a higher penalty is imposed on infeasible solutions. As a consequence, in the early stages of the search process, highly infeasible solutions are also admissible, and therefore, the search agents are allowed to more freely explore the search space, but as the search progresses, feasible solutions are preferred over infeasible ones [50]. If the i-th frequency constraint is violated, then the value of c_i depends on the severity of the violation; otherwise, it is set to 0. This can be expressed mathematically as follows:

$$c_i = \begin{cases} \left| 1 - \dfrac{\omega_i}{\omega_i^*} \right|, & \text{if the } i-\text{th frequency constraint is violated} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

Consequently, the optimization problem can be rewritten as follows:

$$\text{Minimize: } p(\boldsymbol{A}) = f(\boldsymbol{A}) \times f_{penalty}(\boldsymbol{A}) \tag{6}$$

where $p(\boldsymbol{A})$ is the penalized objective function.

In order to determine the natural frequencies and vibration modes of a structure with the finite element method, the following algebraic equation should be solved [51]:

$$k\phi_n = \omega_n^2 m\phi_n \tag{7}$$

where m and k are the mass and stiffness matrices of the structure, respectively, and ω_n and ϕ_n are the n-th natural frequency and natural vibration mode of the structure, respectively.

2.2 Overview of arithmetic optimization algorithm (AOA)

The arithmetic optimization algorithm (AOA) is a population-based metaheuristic optimization algorithm developed by Abualigah et al. in 2021 [39]. Arithmetic is a branch of mathematics concerned with numbers and their addition, subtraction, multiplication, and division [52]. Accordingly, AOA simulates the distribution characteristics of the four basic arithmetic operations of addition (*A*), subtraction (*S*), multiplication (*M*), and division (*D*). Like other metaheuristic algorithms, the search process of AOA can be divided into two main phases: exploration and exploitation. In the exploration phase, the position of the search agents (candidate solutions) are updated based on multiplication and division operators, whereas the exploitation phase deals with addition and subtraction operators. Fig. 2 shows the order of arithmetic operations and their supremacy from outside to inside. In the following subsections, the mathematical formulation of AOA is briefly presented.

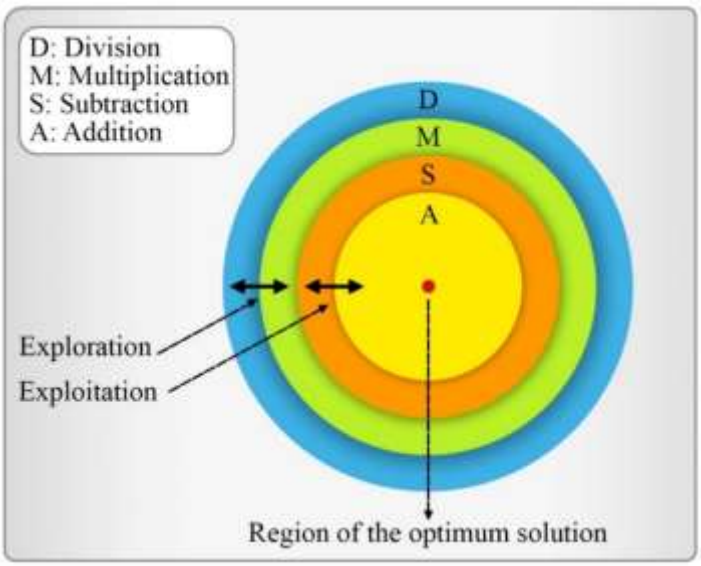

Figure 2. The order of arithmetic operations and their supremacy (from outside to inside) [44]

2.2.1 Initialization phase

Like other population-based metaheuristics, AOA starts with a population of randomly generated candidate solutions (X), as shown in Eq. (1). Note that, whenever the best solution

of the current iteration is better than the best solution found so far, then the best solution found so far is updated.

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & \cdots & x_{1,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i,1} & \cdots & x_{i,j} & \cdots & x_{i,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{N,1} & \cdots & x_{N,j} & \cdots & x_{N,n} \end{bmatrix} \tag{8}$$

where N denotes the number of candidate solutions in the population; n stands for the number of design variables; and $x_{i,j}$ represents the j-th design variables of the i-th candidate solution of the initial population.

The AOA employs a dynamic function, named Math Optimizer Accelerated (MOA), for switching between exploration and exploitation phases during the search process. The MOA function is defined as follows:

$$MOA(C_{Iter}) = Min + C_{Iter} \times \left(\frac{Max - Min}{M_{Iter}}\right) \tag{9}$$

where M_{Iter} is the maximum number of iterations; C_{Iter} is the current iteration number and varies between 1 and M_{Iter}; $MOA(C_{Iter})$ denotes the value of the function MOA at the current iteration; and Min and Max denote the minimum and maximum values of MOA function, respectively. In this work, Min and Max are set to 0.2 and 0.9, respectively, as suggested by Abualigah et al. [39].

When $MOA(C_{Iter}) < r_1$, the search space is explored using the multiplication (*M*) and division (*D*) operators, but when $MOA(C_{Iter}) \geq r_1$, the promising regions of the search space located in the exploration phase are exploited using the addition (*A*) and subtraction (*S*) operators. Note that r_1 is a uniformly distributed pseudorandom number between 0 and 1. As can be observed from Eq. (2), $MOA(C_{Iter})$ starts from $Min + (Max - Min)/M_{Iter}$ at the first iteration (i.e., $C_{Iter} = 1$) and increases linearly until it reaches Max at the last iteration (i.e., $C_{Iter} = M_{Iter}$). As a result, $MOA(C_{Iter})$ can smoothly switch between the exploration and exploitation phases.

2.2.2 Exploration phase

As aforementioned, arithmetic operations of multiplication (*M*) and division (*D*) are employed in the exploration phase of AOA to guide the exploration of the search space. Because of the highly scattering nature of *M* and *D* operators, diverse regions of the search space could be explored. These operators are not able to thoroughly exploit the promising regions of the search space. However, they are required to ensure that all the search space is explored enough to provide a reliable estimate of the global optimum. The exploration phase of the AOA is executed based on the following position update rule:

$$x'_{i,j} = \begin{cases} x_{Best,j} \div (MOP + \varepsilon) \times \Big(\big(UB_j - LB_j \big) \times \mu + LB_j \Big), & r_2 > 0.5 \\ x_{Best,j} \times MOP \times \Big(\big(UB_j - LB_j \big) \times \mu + LB_j \Big), & \text{otherwise} \end{cases} \tag{10}$$

where $x'_{i,j}$ is the updated value of the j-th design variable of the i-th candidate solution; $x_{Best,j}$ denotes the j-th design variables the best solution found so far; LB_j and UB_j are the lower and upper bounds of the j-th design variables, respectively; ε is a very small floating-point number (i.e., 2^{-52}) to avoid singularity; r_2 is a uniformly distributed pseudorandom number between 0 and 1; and μ is a parameter to control the search process, which is set to 0.5 as suggested by Abualigah et al. [39]. Also, MOP denotes a function called Math Optimizer Probability, which is established by the following equation:

$$MOP(C_{Iter}) = 1 - \left(\frac{C_{Iter}}{M_{Iter}} \right)^{1/\alpha} \tag{11}$$

where $MOP(C_{Iter})$ denotes the value of the coefficient *MOP* at the current iteration number, and α is a sensitive parameter reflecting the accuracy of exploitation over the iterations, which is set to 5 as suggested by Abualigah et al. [39].

As can be observed from Eq. (3), the division (*D*) operator is conditioned by $r_2 > 0.5$ and the multiplication (*M*) operator will be ignored until the division (*D*) operator completes its search task; otherwise, the multiplication (*M*) operator is adopted to implement the search process instead of the *D*.

2.2.3 Exploitation phase

In contrast with multiplication (*M*) and division (*D*) operators that produce large step sizes which result in highly scattered population of candidate solutions, addition (*A*) and subtraction (*S*) operators produce small step sizes which lead to highly dense population of candidate solutions. As a result, these operators can easily intensify the search in the promising regions of the search space detected in the exploration phase. The exploitation phase of the AOA is performed based on the following position update rule:

$$x'_{i,j} = \begin{cases} x_{Best,j} - MOP \times \Big(\big(UB_j - LB_j \big) \times \mu + LB_j \Big), & r_3 > 0.5 \\ x_{Best,j} + MOP \times \Big(\big(UB_j - LB_j \big) \times \mu + LB_j \Big), & \text{otherwise} \end{cases} \tag{12}$$

where r_3 is a uniformly distributed pseudorandom number between 0 and 1.

From Eq. (3), it can be seen that the subtraction (*S*) operator is conditioned by $r_3 > 0.5$ and the addition (*A*) operator will be neglected until the subtraction (*S*) operator performs its search task; otherwise, the Addition (*A*) operator is employed to guide the search process instead of the *S*.

It is noted that, in both exploration and exploitation phases, if the penalized objective function value corresponding to the updated position of the i-th candidate solution is better than that of its current position, the new position will replace its corresponding current

position. For a minimization optimization problem, this can be expressed mathematically as follows:

$$x_i = \begin{cases} x_i', & p(x_i') < p(x_i) \\ x_i, & \text{otherwise} \end{cases} \tag{13}$$

where $p(x_i)$ denotes the penalized objective function value of the i-th candidate solution.

The pseudocode of the original AOA is summarized in Fig. 3 as follows [44]:

Initialization phase
Initialize the algorithm parameters: N, Min, Max, M_{Iter}, μ, and α
Generate the initial population of solutions randomly: $X = [x_1; x_2; \ldots; x_N]$
Identify the best solution of initial population: x_{Best}
$C_{Iter} = 1$
while ($C_{Iter} \leq M_{Iter}$) **do**
  $MOA(C_{Iter}) = Min + C_{Iter} \times ((Max - Min)/M_{Iter})$
  $MOP(C_{Iter}) = 1 - (C_{Iter}/M_{Iter})^{(1/\alpha)}$
  for $i = 1$ **to** N **do**
    for $j = 1$ **to** n **do**
      if $MOA(C_{Iter}) < r_1$ **then**
        if $r_2 > 0.5$ **then (Exploration phase)**
          $x'_{i,j} = x_{Best,j} \div MOP \times ((UB_j - LB_j) \times \mu + LB_j)$
        else
          $x'_{i,j} = x_{Best,j} \times MOP \times ((UB_j - LB_j) \times \mu + LB_j)$
        end if
      else
        if $r_3 > 0.5$ **then (Exploitation phase)**
          $x'_{i,j} = x_{Best,j} - MOP \times ((UB_j - LB_j) \times \mu + LB_j)$
        else
          $x'_{i,j} = x_{Best,j} + MOP \times ((UB_j - LB_j) \times \mu + LB_j)$
        end if
      end if
    end for
    Enforce design variable constraints
    if $p(x_i') < p(x_i)$ **then**
      $x_i = x_i'$
    end if
    Update the best solution found so far: x_{Best}
  end for
  $C_{Iter} = C_{Iter} + 1$
end while
return x_{Best}

Figure 3. Pseudocode of the original AOA [44]

2.3 Improved AOA (IAOA)

The original AOA represents an interesting source in inspiration for exploring the search space. However, as observed in previous studies, the original AOA suffers from premature

stagnation in non-optimal solutions [42-43]. This may be due to the rapid loss of population diversity in the early stages of the search process, caused by poor exploration of the search space, as noted by Kaveh and Biabani Hamedani [44]. In order to overcome the shortcomings mentioned below for the original AOA, Kaveh and Biabani Hamedani [44] have recenly proposed an improved version of AOA, called IAOA, and applied it to discrete structural optimization problems. In the following, after a detailed explanation of the shortcomings of the original AOA noted by Kaveh and Biabani Hamedani [44], we review the IAOA method in detail. The exploration and exploitation phases of the original AOA are clearly distinguished from each other by their unique position updating rules (i.e., Eqs. (10) and (12) for exploration and exploitation phases, respectively) [44]. Exploration means the ability to generate diverse solutions so as to explore the search space globally [53]. However, it can be deduced from Eq. (10) that the exploration phase of the AOA consists of focusing the search around the best solution found so far, which may rapidly reduce the population diversity in the early stages of the search process [44]. As a result, the original AOA suffers from the problem of poor exploration and may converge prematurely to non-optimal solutions [44]. On the other hand, the position updating rule given by Eq. (10) involves the bounds of design variables, which means that the exploration phase of the original AOA depends on the bounds of design variables [44]. This may cause difficulties in convergence, especially when bounds of the design variables are too narrow or too wide [44]. In fact, when the bounds of the design variables are too conservative, large step sizes are generated for the movement of solutions in the search space, which may result in exceeding the boundary of the search space [44]. On the other hand, when the lower and upper bounds are very close to each other, small step sizes are generated for the movement of solutions in the search space, which may lead to premature convergence to non-optimal solutions [44]. In addition, in the case where the design variables have the same lower and upper bounds, the original AOA suffers from another serious drawback which will be discussed below in more detail [44].

Let us consider the case where all design variables have the same bounds (i.e., $UB_1 = UB_2 = \cdots = UB_n$ and $LB_1 = LB_2 = \cdots = LB_n$). This is the case of discrete structural optimization where the design variables are selected from a predefined set of standard sections. In such cases, it follows from Eq. (11) that, in each iteration, when $r_2 > 0.5$, all the design variables of the best solution found so far (i.e., $x_{Best,j},\ j = 1,2,\dots,n$) are scaled by the same factor $(MOP + \varepsilon) \times ((UB_j - LB_j) \times \mu + LB_j)$. Similarly, in each iteration, when $r_2 \leq 0.5$, all the design variables of the best solution found so far are multiplied by the same factor $MOP \times ((UB_j - LB_j) \times \mu + LB_j)$. As a result, in the exploration phase of each iteration of the original AOA, all the design variables of the best solution found so far are scaled (i.e., multiplied or divided) by only two factors, which leads to the exploration of a very limited region of the search space. This may cause the loss of population diversity, which may result in slow convergence and premature convergence to non-optimal solutions [44]. This is also true for the exploitation phase of the original AOA (see Eq. (12)), which will be discussed later.

In order to address the drawbacks of the exploration phase of the original AOA, on the basis of division and multiplication operators, the position updating rule of the exploration phase of the IAOA has been proposed as follows [44]:

$$x'_{i,j} = \begin{cases} x_{i,j} \div \left(1 + (-1)^{randi([1,2])} \times 0.5 \times rand \times \overline{MOP}\right), & r_2 > 0.5 \\ x_{i,j} \times \left(1 + (-1)^{randi([1,2])} \times 0.5 \times rand \times \overline{MOP}\right), & \text{otherwise} \end{cases} \quad (14)$$

where $x_{i,j}$ is the current value of the j-th design variable of the i-th candidate solution; $rand$ is a uniformly distributed pseudorandom number between 0 and 1; $randi([1,2])$ returns a pseudorandom scalar integer between 1 and 2; and $\overline{MOP}$ is a parameter-free version of the function MOP, which is defined as follows [44]:

$$\overline{MOP}(C_{Iter}) = \left(1 - \frac{C_{Iter}}{M_{Iter}}\right)^{randi([1,2])} \quad (15)$$

In contrast to the original AOA, the exploration phase of the IAOA implies focusing the search around the current position of solutions, as can be seen from Eq. (14) [44]. In other words, in the exploration phase of the IAOA, the position of each solution is updated with respect to its current position, thus allowing a broader exploration of the search space to avoid diversity loss during the search process [44]. In addition, the random numbers used in Eqs. (14) and (15) cause that different step sizes are generated more the movement of solutions in the search space, which can enhance the exploration and maintain the diversity of the population [44]. It can be seen that Eq. (14) does not depend on the bounds of design variables, which may help to avoid convergence difficulties [44].

As Eq. (12) suggests, the position updating rule of solutions in the exploitation phase of the original AOA, similar to its exploration phase, involves focusing the search around the best solution found so far. As aforementioned, the major drawback of the exploitation phase of the original AOA arises when all the design variables have the same lower and upper bounds. In this case, as Eq. (12) suggests, in each iteration, the same factor $MOP \times ((UB_j - LB_j) \times \mu + LB_j)$ is subtracted from (when $r_3 > 0.5$) or added to (when $r_3 \leq 0.5$) all the design variables of the best solution found so far. As noted earlier, this means that, in each iteration of the exploitation phase of the original AOA, all the design variables of the best solution found so far are scaled (i.e., added or subtracted) by only two factors. As a result, exploitation does not take place effectively. In order to overcome the drawback mentioned above, based on subtraction and addition operators, the following position updating rule has been proposed for the exploitation phase of IAOA [44]:

$$x'_{i,j} = \begin{cases} x_{Best,j} - x_{Best,j} \times rand \times \overline{MOP} \times \left(UB_j - LB_j\right), & r_3 > 0.5 \\ x_{Best,j} + x_{Best,j} \times rand \times \overline{MOP} \times \left(UB_j - LB_j\right), & \text{otherwise} \end{cases} \quad (16)$$

By using Eqs. (15) and (16), different step sizes are generated for the movement of solutions in the search space, thus allowing a better exploitation of the best solution found so far. The formulation of the original AOA involves the four algorithm-specific parameters of Min, Max, α, and μ that should be tuned for any specific application. From Eqs. (14), (15), and (16), it is seen that the parameters α and μ are eliminated from the formulation of the IAOA, which makes it easier to be implemented compared with the original AOA. To the best of our knowledge, this is the first attempt to apply both the AOA and IAOA for

structural optimization problems with frequency constraints. The pseudocode and flowchart of IAOA are shown in Figs. 4 and 5, respectively.

Initialization phase
Initialize the algorithm parameters: N, Min, Max, and M_{Iter}
Generate the initial population of solutions randomly: $X = [x_1; x_2; \ldots; x_N]$
Identify the best solution of initial population: x_{Best}
$C_{Iter} = 1$
while $(C_{Iter} \leq M_{Iter})$ **do**
  $MOA(C_{Iter}) = Min + C_{Iter} \times ((Max - Min)/M_{Iter})$
  for $i = 1$ **to** N **do**
    for $j = 1$ **to** n **do**
      $\overline{MOP}(C_{Iter}) = (1 - C_{Iter}/M_{Iter})^{randi([1,2])}$
      if $MOA(C_{Iter}) < r_1$ **then**
        if $r_2 > 0.5$ **then (Exploration phase)**
          $x'_{i,j} = x_{i,j} \div \left(1 + (-1)^{randi([1,2])} \times 0.5 \times rand \times \overline{MOP}\right)$
        else
          $x'_{i,j} = x_{i,j} \times \left(1 + (-1)^{randi([1,2])} \times 0.5 \times rand \times \overline{MOP}\right)$
        end if
      else
        if $r_3 > 0.5$ **then (Exploitation phase)**
          $x'_{i,j} = x_{Best,j} - x_{Best,j} \times rand \times \overline{MOP} \times ((UB_j - LB_j))$
        else
          $x'_{i,j} = x_{Best,j} + x_{Best,j} \times rand \times \overline{MOP} \times ((UB_j - LB_j))$
        end if
      end if
    end for
    Enforce design variable constraints
    if $p(x'_i) < p(x_i)$ **then**
      $x_i = x'_i$
    end if
    Update the best solution found so far: x_{Best}
  end for
  $C_{Iter} = C_{Iter} + 1$
end while
return x_{Best}

Figure 4. Pseudocode of the Improved AOA (IAOA) [44]

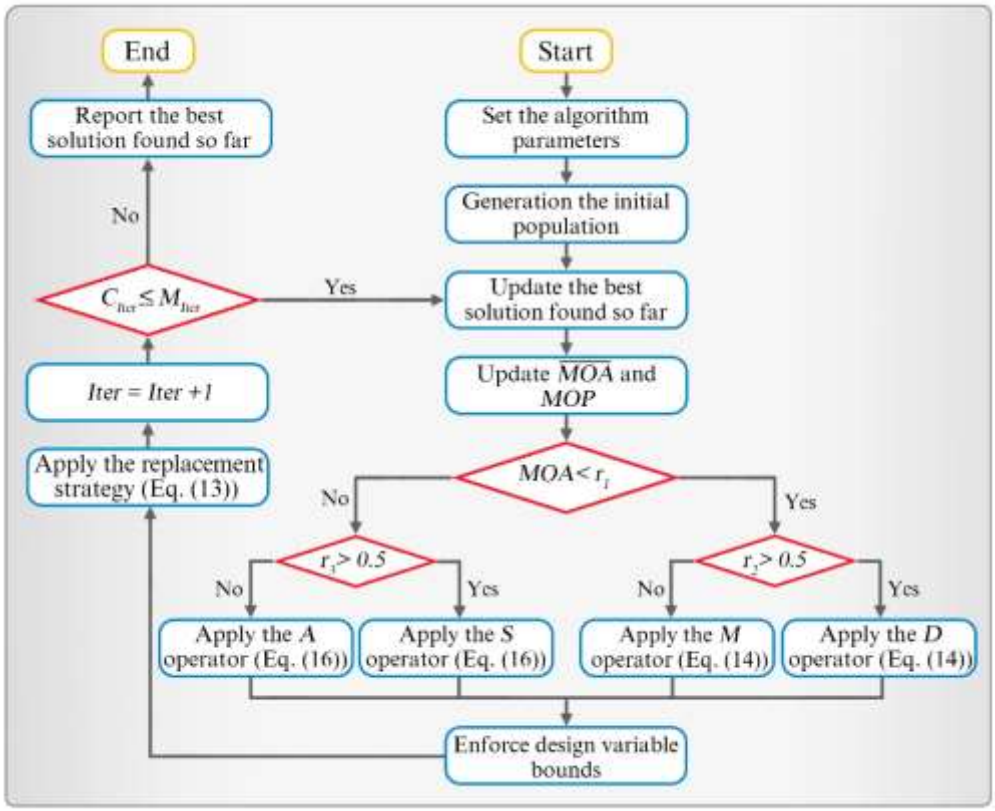

Figure 5. Flowchart of the improved AOA (IAOA) [44]

3. RESULTS AND DISCUSSION

3.1 Numerical examples

In this section, to demonstrate the effectiveness and efficiency of both the AOA and IAOA, three numerical examples of size optimization of large-scale truss structures with frequency constraints are investigated. The results achieved by IAOA are compared with those of the original AOA and some state-of-the-art methods. The examples include a 600-bar single-layer dome-shaped truss with 25 design variables, an 1180-bar single-layer dome-shaped truss with 59 design variables, and a 1410-bar double-layer dome-shaped truss with 47 design variables. Material properties, cross-sectional area bounds, and frequency constraints for all examples are listed in Table 1. In all examples, for both the AOA and IAOA, the population size is to $N = 20$ and the maximum number of iterations is set to $M_{Iter} = 1000$. These values were selected on the basis of the results of preliminary experiments and other similar works in the literature [54]. In addition, the parameters Min, Max, α, and μ are fixed at 0.2, 0.9, 5, and 0.5, as suggested by Abualigah et al. [39]. In order to consider the stochastic nature of the optimization process, ten independent optimization runs were carried out for each example. The statistical results of the original AOA and IAOA are provided in terms of the best, average, worst, and standard deviation

(SD) of the optimized weights obtained over ten independent runs. The maximum number of finite element analyses (Max_{NFE}) of different methods are also provided. The finite element method (FEM) model for free vibration analysis of truss structures and the optimization algorithms were programmed in the Matlab environment. The FEM model was verified against the optimization results reported by Kaveh and Ilchi Ghazaan [55]. The simulations were performed on a PC with Intel(R) Core (TM) i5-7200U CPU 2.50 GHz 2.71 GHz, and 8.00 GB RAM with a Microsoft Windows 10, 64-bit operating system.

Table 1: Optimal results obtained for the complete graphs

Property	600-bar truss	1180-bar truss	1410-bar truss
Elasticity modulus (N/m^2)	2×10^{11}	2×10^{11}	2×10^{11}
Material density (kg/m^3)	7850	7850	7850
Cross-sectional area bounds (m^2)	$0.0001 \le A_i \le 0.01$	$0.0001 \le A_i \le 0.01$	$0.0001 \le A_i \le 0.01$
Frequency constraints (Hz)	$\omega_1 \ge 5, \omega_3 \ge 7$	$\omega_1 \ge 7, \omega_3 \ge 9$	$\omega_1 \ge 7, \omega_3 \ge 9$

3.1.1 The 600-bar dome-like truss

The first example is a 600-bar dome-shaped truss structure shown in Fig. 6(a-b). This is a cyclic symmetric structure composed of 24 identical substructures. The angle between two adjacent substructures is 15°. Each substructure has 9 nodes and 25 elements, as depicted in Fig. 6(c). The nodal coordinates of the first substructure in the Cartesian coordinate system are given in Table 2. The connectivity information of the first substructure is also given in Table 3. The cross-sectional area of each element of the substructure is considered as a sizing design variable. The layout of the structure is not changed during the design process. Therefore, this is an optimization problem with 25 sizing design variables. A non-structural mass of 100 kg is attached to all free nodes of the structure. The material properties, frequency constraints, and cross-sectional area bounds of the structure are listed in Table 1. This problem has been previously studied by many researchers using various metaheuristic techniques, such as CPA [11], CRPSO [56], JA [57], PFJA [58], etc.

In this example, the effectiveness of the modifications made in IAOA will be discussed in detail. Table 3 provides a comparison between the optimal results achieved by the present algorithms with those of the other methods existing in the literature. It is seen that the IAOA achieved the best performance in terms of the best weight, average weight, worst weight, and standard deviation. The IAOA obtained a feasible optimal solution with a structural weight of 6067.73 kg after 19140 FE analyses, which is better than the best weights obtained by all the other algorithms, namely 6132.30 kg for CRPSO [56], 6112.64 kg for JA [57], 6336.85 kg for CPA [11], 6333.25 kg for PFJA [58], 6140.51 kg for ECBO-Cascade [9], and 6416.58 kg for AOA. The worst weight obtained by the IAOA over ten independent runs is 6077.32 kg, which is even better than the average weights of the others. Compared to the original AOA, IAOA showed significant improvements in all aspects. In fact, the AOA showed the worst performance in terms of the best and average weights, which may be

because of the drawbacks mentioned for the AOA. In terms of the convergence speed, the maximum number of finite element (FE) analyses of the IAOA is 20000, which is less than or equal to those of other methods, except the JA [57]. However, the IAOA achieved a feasible optimal solution with a structural weight of 6100.00 kg after 12880 FE analyses, which is better than the best weight of the JA after 15000 FE analyses. Fig. 7 illustrates the diversity of the optimal weights found by AOA and IAOA over ten independent runs. As it is seen from the figure, the AOA converged prematurely to non-optimal solutions in all runs. In contrast, IAOA consistently converged to near-optimal solutions, confirming the high robustness of these methods. The average convergence histories of the AOA and IAOA are also plotted for comparison in Fig. 8. It demonstrates clearly that the IAOA has the fastest convergence rate during the entire optimization process. On the other hand, it is observed that the AOA converged prematurely within the first 10180 FE analyses (i.e., the first 509 iterations) and no improvement was observed in the average optimized weight. This comes from the fact that the original AOA suffers severely from the rapid loss of population diversity and is prone to stagnation in non-optimal solutions. However, thanks to the effective exploration of the search space in the IAOA, it significantly improves the convergence rate. Fig. 9 compares the average number of solution replacements occurred in the AOA and IAOA over ten independent runs. It is noted that the solution replacement means the current position of a solution is replaced by its corresponding updated position. As can be seen from the figure, the average number of solution replacements observed in the IAOA after 20000 FE analyses is about 1365, which is much more than about 234 of the AOA, which can be attributed to a better and more effective search process in the IAOA compared with the original AOA. Table 4 lists the first five natural frequencies corresponding to the optimal designs provided by the AOA, IAOA, and other methods available in the literature. As expected, all of the frequency constraints are satisfied.

Table 2: Nodal coordinates of the first sub-structure of the 600-bar dome-like truss (m)

Node number	Coordinates (x, y, z)
1	(1.0, 0.0, 7.0)
2	(1.0, 0.0, 7.5)
3	(3.0, 0.0, 7.25)
4	(5.0, 0.0, 6.75)
5	(7.0, 0.0, 6.0)
6	(9.0, 0.0, 5.0)
7	(11.0, 0.0, 3.5)
8	(13.0, 0.0, 1.5)
9	(14.0, 0.0, 0.0)

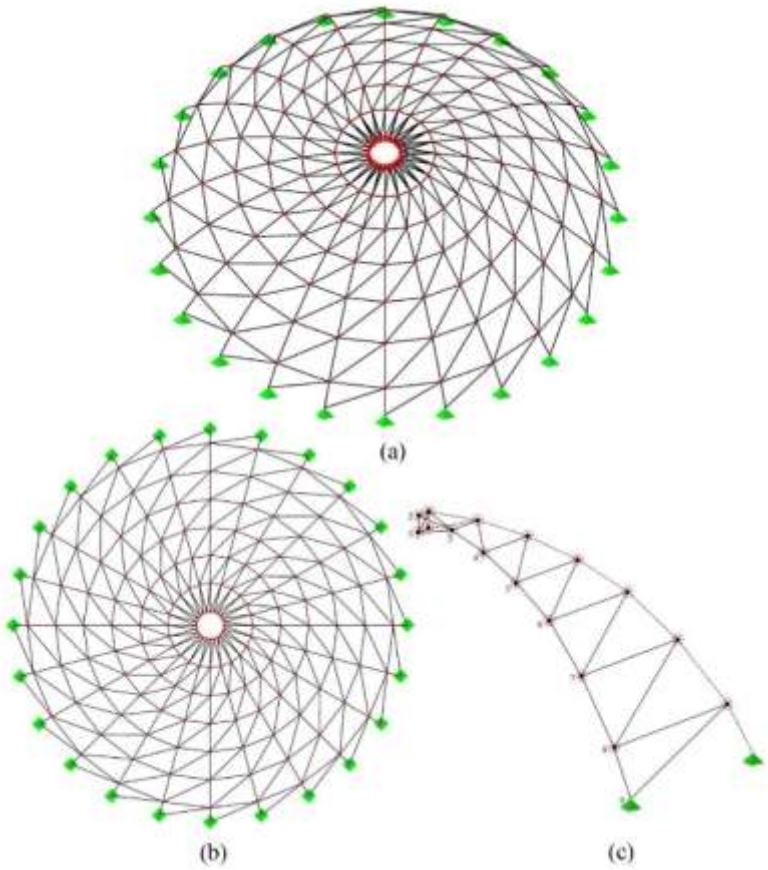

Figure 6. Schematic of the 600-bar dome: (a) perspective view; (b) top view; (c) substructure

Table 3: Optimal results of the 600-bar truss obtained by different algorithms (cm^2)

Element number (element nodes)	CRPSO [56]	JA [57]	CPA [11]	PFJA [58]	ECBO-Cascade [9]	This study	
						AOA	IAOA
1 (1-2)	1.5	1.7518	1.155	1.1867	1.0299	1.3793	1.7980
2 (1-3)	1.5	1.1811	1.304	1.2967	1.3664	1.1927	1.3862
3 (1-10)	7	4.8878	4.178	4.5771	5.1095	3.9187	4.2000
4 (1-11)	1	1.5162	1.335	1.3356	1.3011	5.2569	1.2123
5 (2-3)	16.5	18.1659	18.375	18.3157	17.0572	16.6232	16.5982
6 (2-11)	34.5	36.0764	39.914	38.5097	34.0764	29.5133	38.5906
7 (3-4)	12	12.6571	13.609	13.5917	13.0985	12.7923	12.7886
8 (3-11)	15.5	14.6113	16.470	16.8824	15.5882	15.0625	15.5347
9 (3-12)	10.5	11.3198	14.108	13.8766	12.6889	12.4096	11.8675
10 (4-5)	10	8.4580	10.038	9.5286	10.3314	9.8585	9.5076
11 (4-12)	8.5	8.4285	9.514	9.4218	8.5313	7.1074	8.3020
12 (4-13)	9	9.7321	9.329	9.7643	9.8308	8.5387	9.6386
13 (5-6)	7.5	7.2947	6.938	7.2431	7.0101	7.2116	7.0861
14 (5-13)	5.5	6.1922	5.545	5.3913	5.2917	8.6170	5.2601

15 (5-14)	6.5	6.4395	6.763	6.7468	6.2750	5.1681	6.5093
16 (6-7)	5.5	5.4760	5.209	5.1493	5.4305	5.7574	5.5456
17 (6-14)	5	3.2695	3.842	3.8342	3.6414	4.1665	3.5337
18 (6-15)	7.5	8.3724	8.112	8.0665	7.2827	8.0542	7.4756
19 (7-8)	4.5	4.4987	4.252	4.2800	4.4912	5.8614	4.3597
20 (7-15)	2	2.2197	2.227	2.2509	1.9275	2.0260	2.0899
21 (7-16)	4.5	4.6162	4.582	4.5372	4.6958	4.8432	4.4251
22 (8-9)	4	3.0667	3.336	3.5615	3.3595	4.8705	3.4614
23 (8-16)	2	1.8549	1.725	1.7744	1.7067	1.6097	1.9229
24 (8-17)	4.5	4.7960	4.675	4.6445	4.8372	6.6149	4.8054
25 (9-17)	1.5	1.6029	1.673	1.6141	2.0253	1.6336	1.5993
Best weight (kg)	6132.30	6112.64	6336.85	6333.25	6140.51	6416.58	6067.51
Average weight (kg)	6682.32	6146.19	6376.01	6380.31	6175.33	6935.54	6072.83
Worst weight (kg)	10409.94	6783.48	-	-	-	7775.88	6077.32
SD (kg)	999.25	17.24	90.39	47.40	34.08	447.00	3.10
Max_{NFE}	20000	15000	40000	25000	20000	20000	20000

Table 4: The first five natural frequencies of the 600-bar dome found by different methods (Hz)

Frequency number	CRPSO [56]	JA [57]	CPA [11]	PFJA [58]	ECBO-Cascade [9]	This study	
						AOA	IAOA
1	5.0231	5.0804	5.000	5.0011	5.001	5.0006	5.0000
2	-	5.0804	5.000	5.0011	5.001	5.0006	5.0000
3	7.0013	7.0001	7.000	7.0000	7.001	7.0034	7.0000
4	-	7.0001	7.000	7.0000	7.001	7.0034	7.0000
5	-	7.0006	7.000	7.0000	7.002	7.0480	7.0000

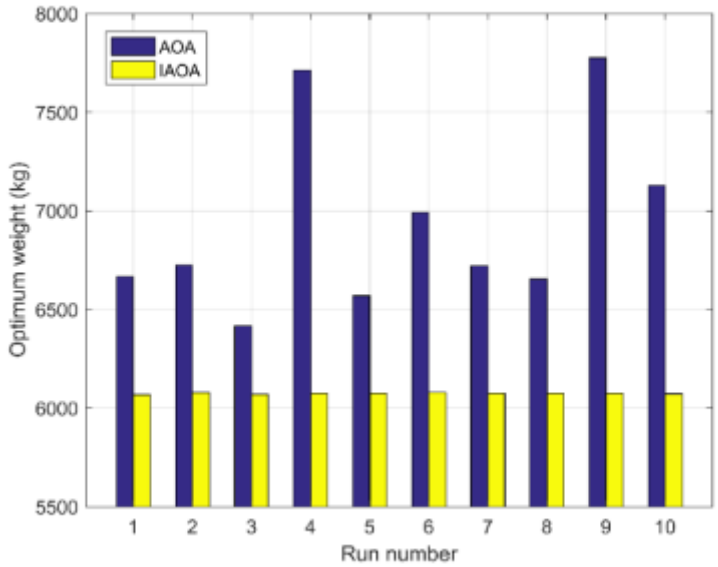

Figure 7. Diversity of the optimal weights of the 600-bar truss found by different algorithms

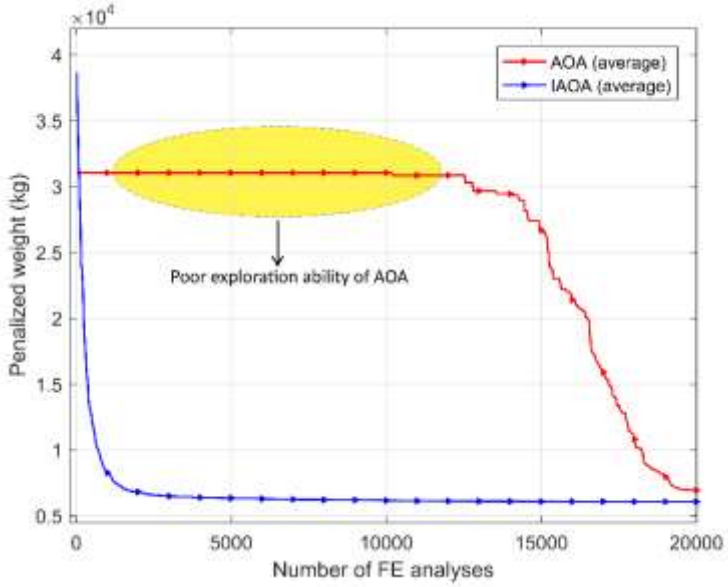

Figure 8. The weight convergence histories of the 600-bar truss obtained by different algorithms

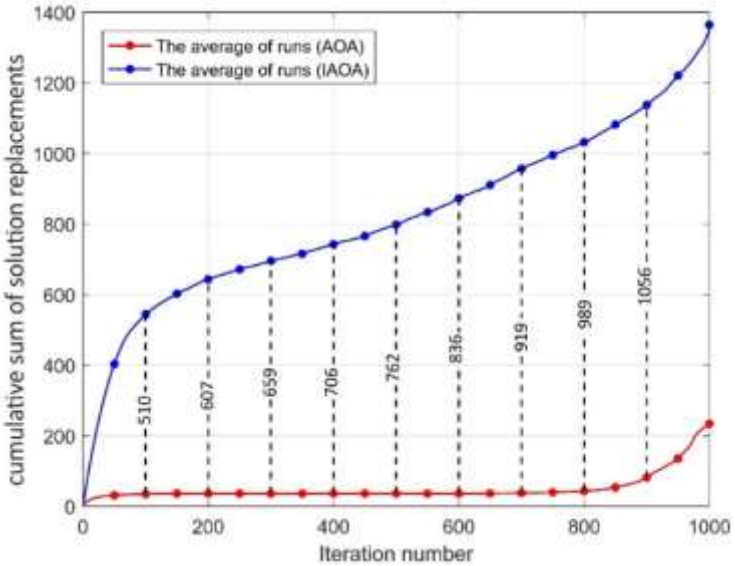

Figure 9. The number of solution replacements observed in the AOA and IAOA (600-bar truss)

3.1.2 The 1180-bar dome-like truss

The 1180-bar dome-shaped truss shown in Fig. 10(a-b) is considered as the second example. This is also a cyclic symmetric structure consisted of 20 identical substructures. The angle between two adjacent substructures is 18°. Each substructure is composed of 20 nodes and 59 elements, as shown in Fig. 10(c). The nodal coordinates of the first substructure in the Cartesian coordinate system are presented in Table 5. The connectivity information of the first substructure is also given in Table 6. The cross-sectional areas of the elements of the

substructure is considered as sizing design variables. The layout of the structure remains unchanged during the design procedure. Accordingly, this is a sizing optimization problem with 59 design variables. Similar to the previous example, a non-structural mass of 100 kg is attached to each free node of the structure. Table 1 lists the material properties, frequency constraints, and cross-sectional area bounds of the structure. This problem has been previously addressed in the literature using different metaheuritic techniques, such as JA [59], CPA [11], PFJA [58], etc.

Table 6 compares the optimal solutions found by the AOA and IAOA with those of other methods in the literature. The results suggest that the IAOA obtained the best results in terms of the average weight and worst weight among all other methods, while the original AOA showed the worst performance in all aspects. The best weights found by the IAOA is 37386.45 kg, which is better than the best weights of the other methods. Compared to the original AOA, IAOA provided significantly better results in all investigated parameters. The IAOA obtained a feasible optimal solution with a structural weight of 92249.08 kg after only 600 FE analyses, which is better than that the best weight found by the AOA after 20000 FE analyses. In terms of the convergence rate, the maximum number of FE analyses of the IAOA is 20000, which is less than or equal to those of the others. It is obvious from Table 6 that the AOA performed significantly worse than all other methods, confirming that the original AOA suffers from different problems. The diversity of the optimal weights obtained by AOA and IAOA over ten independent runs are depicted in Fig. 11. It is easily seen that, in all the ten runs, the original AOA converged prematurely to non-optimal solutions, indicating the problem of premature convergence to non-optimal solutions in the AOA. In contrast, the IAOA consistently converged to solutions near the global minimum. Fig. 12 compares the average convergence histories of the AOA and IAOA. It reveals that the IAOA converges faster than AOA. In addition, it is seen that, similar to the previous example, the AOA converged prematurely within the first 18040 FE analyses (i.e., the first 902 iterations) and no improvement was observed in the average optimized weight. This is due to poor exploration of the search space in the original AOA, leading to the rapid loss of population diversity and, thus, premature convergence to non-optimal solutions. Table 7 provides the first five natural frequencies evaluated at the optimal designs obtained by the AOA, IAOA, and other methods in the literature. None of the frequency constraints are violated, as expected.

Table 5: Nodal coordinates of the first sub-structure of the 1180-bar dome-like truss (m)

Node number	Coordinates (x, y, z)	Node number	Coordinates (x, y, z)
1	(3.1181, 0.0, 14.6723)	11	(4.5788, 0.7252, 14.2657)
2	(6.1013, 0.0, 13.7031)	12	(7.4077, 1.1733, 12.9904)
3	(8.8166, 0.0, 12.1354)	13	(9.9130, 1.5701, 11.1476)
4	(11.1476, 0.0, 10.0365)	14	(11.9860, 1.8984, 8.8165)
5	(12.9904, 0.0, 7.5000)	15	(13.5344, 2.1436, 6.1013)
6	(14.2657, 0.0, 4.6358)	16	(14.4917, 2.2953, 3.1180)
7	(14.9179, 0.0, 1.5676)	17	(14.8153, 2.3465, 0.0)
8	(14.9179, 0.0, -1.5677)	18	(14.9179, 2.2953, -3.1181)
9	(14.2656, 0.0, -4.6359)	19	(13.5343, 2.1436, -6.1014)
10	(12.9903, 0.0, -7.5001)	20	(3.1181, 0.0, 13.7031)

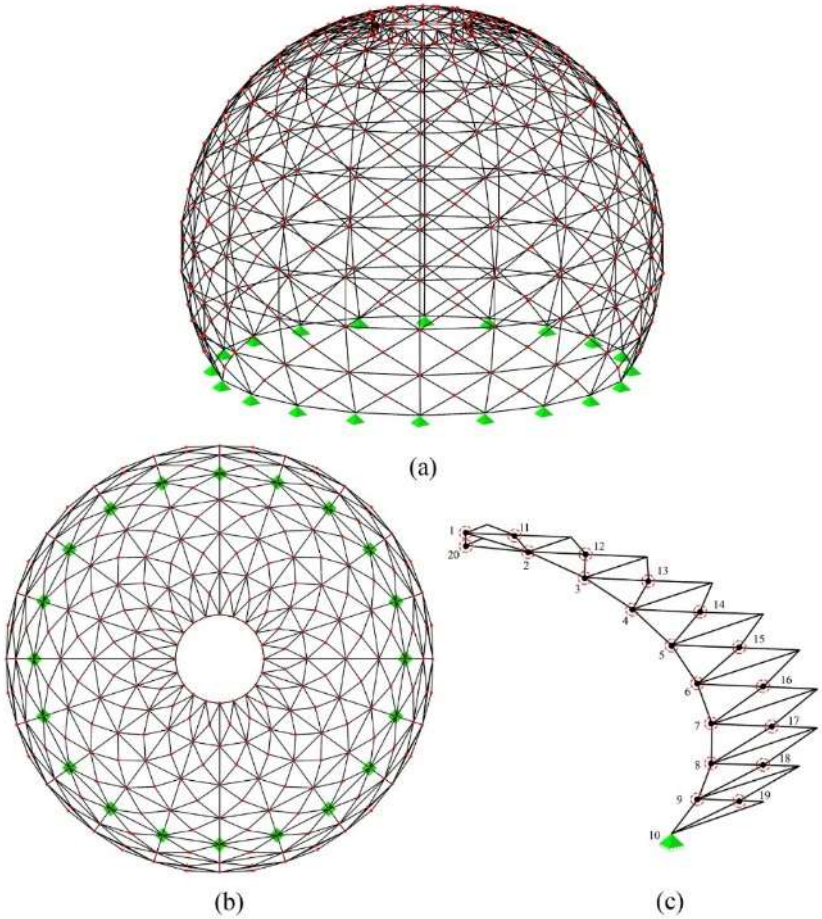

Figure 10. Schematic of the 1180-bar dome: (a) perspective view; (b) top view; (c) substructure

Table 6: Optimal results of the 1180-bar truss obtained by different algorithms (cm^2)

Element number (element nodes)	Chaotic WSA [60]	ECBO-Cascade [9]	PFJA [58]	CPA [11]	JA [59]	This study	
						AOA	IAOA
1 (1-2)	6.9078	8.0110	7.952	7.877	7.2951	25.3850	7.4342
2 (1-11)	10.7524	8.7028	10.466	11.025	10.0202	44.9939	9.4269
3 (1-20)	2.9439	3.1616	2.089	3.325	2.2254	23.1754	2.4849
4 (1-21)	13.487	13.6820	14.219	14.672	14.4745	59.7622	14.4985
5 (1-40)	3.3147	3.2865	3.944	2.894	3.1635	17.4938	3.3486
6 (2-3)	6.9318	6.0397	5.979	5.872	6.1055	22.2250	6.1802
7 (2-11)	7.6325	8.4370	7.775	7.026	7.4452	31.2942	6.9541
8 (2-12)	6.2343	6.4122	6.351	6.746	6.1321	4.5063	6.8006
9 (2-20)	1.3899	2.6346	1.896	1.608	2.0210	16.5124	1.7084
10 (2-22)	12.9919	11.7440	11.908	11.696	11.6685	12.0677	12.5068
11 (3-4)	6.9162	7.9272	7.241	7.523	6.7546	13.0891	6.7561
12 (3-12)	5.119	5.4548	5.647	6.162	5.6377	9.6147	5.4502

13 (3-13)	8.7795	6.7221	6.700	6.769	7.0624	14.0659	6.7871
14 (3-23)	6.684	8.1544	7.799	7.671	7.0211	13.2000	7.1903
15 (4-5)	9.317	9.7560	9.198	8.732	9.1227	23.5175	8.9336
16 (4-13)	6.483	6.5905	6.282	5.841	5.6027	14.4125	5.8181
17 (4-14)	8.2833	7.0392	7.695	8.545	7.5350	43.0417	7.5646
18 (4-24)	8.0703	6.9219	7.520	7.386	7.4734	5.6947	7.3284
19 (5-6)	12.7141	11.6919	11.840	12.668	12.9880	16.1875	11.9311
20 (5-14)	7.0934	9.8890	7.230	8.733	7.3120	83.8732	7.9595
21 (5-15)	10.069	9.3316	10.211	9.037	10.0703	13.0112	11.4983
22 (5-25)	9.7217	9.1093	9.252	8.903	9.4058	97.1872	8.9429
23 (6-7)	17.2315	18.1212	17.222	17.013	17.2086	72.8158	17.3905
24 (6-15)	9.7761	10.6725	11.417	10.048	10.6750	16.9657	10.6771
25 (6-16)	13.2779	13.5340	14.196	14.057	12.5926	14.1129	13.4863
26 (6-26)	11.5212	12.0248	11.639	12.154	11.6369	57.4867	11.0536
27 (7-8)	21.2086	23.1245	24.065	24.024	21.4858	90.6372	24.5693
28 (7-16)	13.0618	15.2630	13.377	14.143	13.5946	36.0220	14.1344
29 (7-17)	17.9725	18.3075	16.469	17.894	17.4000	21.2952	17.8450
30 (7-27)	14.0147	15.2361	16.057	15.276	14.5421	17.3385	15.3335
31 (8-9)	33.5273	40.0749	34.125	36.006	35.4919	94.4774	34.3383
32 (8-17)	20.1075	18.4775	18.866	18.409	18.6342	24.6606	17.8344
33 (8-18)	23.098	26.0689	24.600	22.644	25.3160	77.7791	24.4304
34 (8-28)	22.0597	21.2213	21.103	22.121	22.2539	43.1877	21.0955
35 (9-10)	49.187	46.3724	47.696	48.973	51.4201	79.7429	49.7797
36 (9-18)	26.835	23.6689	27.760	25.396	24.9577	42.8752	24.5117
37 (9-19)	31.4569	35.0703	33.518	33.599	34.5291	71.3263	34.1196
38 (9-29)	30.2512	27.9369	31.773	31.724	32.2597	72.9074	31.9680
39 (10-19)	34.4764	34.2912	33.592	36.419	35.6710	97.2279	37.3880
40 (10-30)	1.1023	1.0726	1.000	1.004	1.0445	1.3381	1.2415
41 (11-21)	9.9842	8.5106	9.455	10.624	9.6966	28.9967	9.8029
42 (11-22)	7.5443	6.8664	7.189	6.909	7.0081	59.1603	7.1772
43 (12-22)	7.6993	5.8229	6.767	5.644	6.5742	15.8988	6.1189
44 (12-23)	6.0238	5.3986	6.322	5.301	5.6216	32.0946	5.6467
45 (13-23)	6.4087	8.0669	6.720	7.370	7.1827	17.8620	6.3340
46 (13-24)	6.4428	6.9797	6.425	6.301	6.1849	2.9031	6.0740
47 (14-24)	8.4235	7.2735	8.451	7.853	7.7539	17.5726	8.4845
48 (14-25)	8.7143	9.1827	8.176	7.987	7.6625	15.4899	7.8935
49 (15-25)	9.8677	10.6227	10.069	10.460	9.8545	16.2250	10.9118
50 (15-26)	11.4715	11.5740	12.219	11.446	10.4573	11.4122	10.7207
51 (16-26)	15.248	15.5194	13.257	13.918	15.2639	16.8066	13.3498
52 (16-27)	12.9199	14.1342	13.782	14.694	13.1448	48.6628	14.7147
53 (17-27)	19.5895	17.1612	17.573	18.227	16.1681	85.8306	18.4999
54 (17-28)	20.1524	19.0798	19.909	19.235	18.1630	18.3446	17.8163
55 (18-28)	24.519	23.4414	24.019	24.421	23.4467	24.8469	23.7437
56 (18-29)	25.838	26.5726	27.701	22.866	27.4008	33.8253	24.3389
57 (19-29)	36.4546	33.4104	32.918	34.344	34.1134	82.8196	31.4950
58 (19-30)	37.7461	37.1198	37.001	36.964	35.3212	20.2882	36.5415
59 (20-40)	3.7146	4.7593	3.864	5.269	4.1362	1.2654	4.4240

Best weight (kg)	37642.38	37770.71	37695.59	37739.57	37439.44	92764.64	37386.45
Average weight (kg)	37795.53	37885.15	37755.05	37813.34	37453.56	115267.41	37424.50
Worst weight (kg)	-	-	-	-	-	138039.84	37468.30
SD (kg)	165.32	133.84	58.025	61.50	5.91	11932.91	26.66
Max_{NFE}	30000	20000	25000	80000	60000	20000	20000

Table 7: The first five natural frequencies of the 1180-bar dome found by different methods (Hz)

Frequency number	Chaotic WSA [60]	ECBO-Cascade [9]	PFJA [58]	CPA [11]	JA [59]	This study	
						AOA	IAOA
1	7.0001	7.000	7.0000	7.000	7.0001	7.0183	7.0000
2	-	7.001	7.0000	7.000	7.0001	7.0183	7.0000
3	9.0049	9.002	9.0024	9.000	9.0002	9.6930	9.0000
4	-	9.002	9.0024	9.000	9.0002	9.6930	9.0000
5	-	9.062	9.0129	9.000	9.0259	10.1564	9.0149

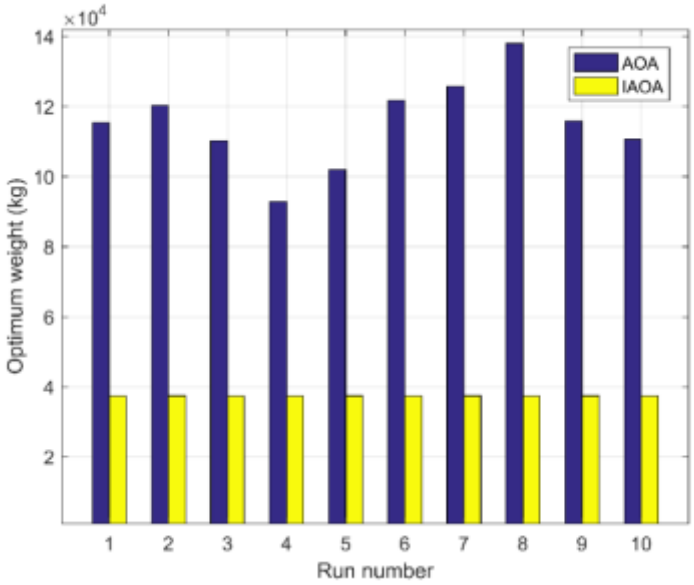

Figure 11. Diversity of the optimal weights of the 1180-bar truss found by different methods

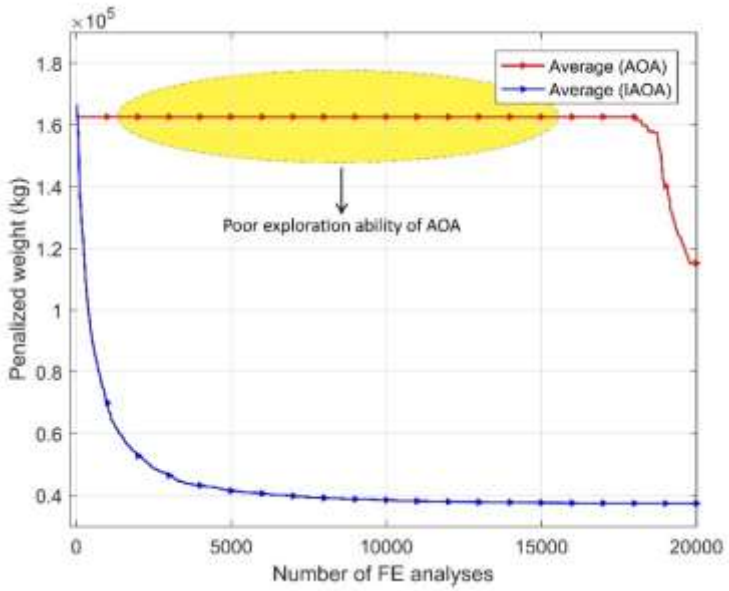

Figure 12. The weight convergence histories of the 1180-bar truss obtained by different methods

3.1.3 The 1410-bar dome-like truss

The last example is a 1410-bar double-layer dome shown in Fig. 13(a-b). As shown in the figure, the dome is a cyclic symmetric structure composed of 30 identical substructures. The angle between two adjacent substructures is 12°. Each substructure consists of 13 nodes and 47 elements, as shown in Fig. 13(c). Table 8 gives the Cartesian coordinates of the nodes of the first substructure. The connectivity information between the nodes of the first substructure is also provided in Table 9. The cross-sectional area of each element of the substructure represents a sizing design variable. In addition, similar to the previous example, the layout of the structure remains unchanged during the design process. Thus, this is a size optimization problem with 47 design variables. A non-structural mass of 100 kg is also attached to all free nodes of the dome. The material properties, frequency constraints, and cross-sectional area bounds of the problem are given in Table 1. This problem has been previously addressed in the literature using different metaheuritic techniques, such as JA and ST-JA [50], CPA [11], PFJA [58], ECBO-Cascade [9], etc.

Table 9 compares the optimal results obtained by the original AOA and IAOA with those of other methods reported in the literature. As expected, the IAOA showed the best performance in terms of the best weight, average weight, and worst weight. The IAOA obtained a feasible optimal solution with a structural weight of 10281.06 kg after 17980 FE analyses, which is better than the best weights found by the other methods. The average weight obtained by the IAOA over ten independent runs is 10303.99 kg, which is even better than the best weights of the other methods, except the ST-JA [50]. Furthermore, the maximum number of FE analyses of the IAOA is 20000, which is lower than or equal to those of the other reported methods. On the other hand, it is easily seen that the AOA showed the worst performance in terms of the best weight, average weight, and worst weight. Fig. 14 shows the diversity of the optimal weights obtained by the JA and its set-theoretical variant [50], AOA, and IAOA. It is apparent that, in all the runs, the original

AOA converged prematurely to non-optimal solutions. However, the ST-JA and IAOA consistently converged to near-optimal solutions. Additionally, the weight convergence histories of the original AOA, IAOA, and JA and its set-theoretical variant [50] are plotted in Fig. 15. It is clear that the IAOA converges faster than the other three methods. Furthermore, it is seen that the AOA converged prematurely within the first 538 iterations (i.e., the first 10760 FE analyses) and no improvement was observed in the average optimized weight. As discussed earlier, this is because of the problem of premature convergence to non-optimal solutions in the AOA. Table 10 shows the first five natural frequencies corresponding to the optimal designs obtained by the AOA, IAOA, and other algorithms in the literature. As expected, there are no any violated frequency constraints.

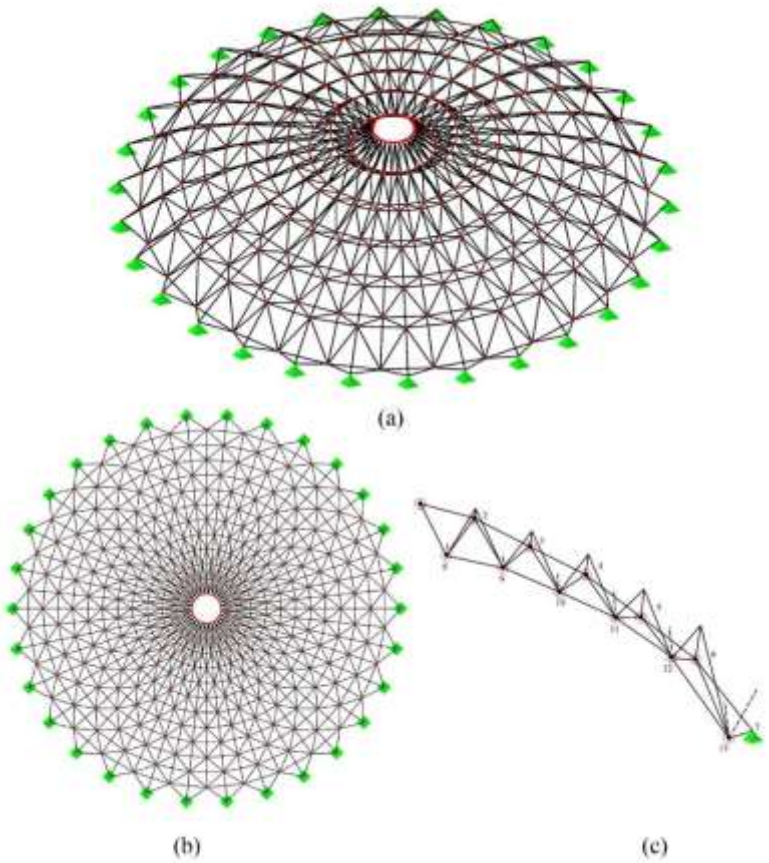

Figure 13. Schematic of the 1410-bar dome: (a) perspective view; (b) top view; (c) substructure

Table 8: Nodal coordinates of the first sub-structure of the 1410-bar dome-like truss (m)

Node number	Coordinates (x, y, z)	Node number	Coordinates (x, y, z)
1	(1.0, 0.0, 4.0)	8	(1.989, 0.209, 3.0)
2	(3.0, 0.0, 3.75)	9	(3.978, 0.418, 2.75)
3	(5.0, 0.0, 3.25)	10	(5.967, 0.627, 2.25)

4	(7.0, 0.0, 2.75)	11	(7.956, 0.836, 1.75)
5	(9.0, 0.0, 2.0)	12	(9.945, 1.0453, 1.0)
6	(11.0, 0.0, 1.25)	13	(11.934, 1.2543, -0.5)
7	(13.0, 0.0, 0.0)		

Table 9: Optimal results of the 1410-bar truss obtained by different algorithms (cm^2)

Element number (element nodes)	CPA [11]	ECBO-Cascade [9]	PFJA [58]	JA [50]	ST-JA [50]	This study	
						AOA	IAOA
1 (1-2)	7.416	7.9969	6.1902	5.1377	5.3860	1.0000	6.3863
2 (1-8)	4.768	6.1723	4.4036	3.7009	4.4361	1.8557	4.9227
3 (1-14)	38.993	35.5011	31.2253	33.9653	27.7395	3.6858	33.6911
4 (2-3)	8.966	10.2510	8.4715	9.5992	8.1780	10.9872	8.7943
5 (2-8)	4.511	5.3727	4.8590	5.5482	6.1189	6.8618	5.7958
6 (2-9)	1.544	1.3488	1.5759	3.6969	1.5996	1.2054	1.2595
7 (2-15)	8.371	11.4427	12.9451	21.8390	18.7495	73.8061	14.3463
8 (3-4)	9.276	9.7157	9.3263	8.6255	9.2413	4.9589	8.7474
9 (3-9)	3.583	1.3005	3.2716	3.5731	2.6310	2.6653	2.0556
10 (3-10)	3.476	2.5046	3.2878	3.3162	2.2419	1.1121	2.5067
11 (3-16)	15.531	10.7849	12.6719	9.0269	7.9773	3.1016	9.1437
12 (4-5)	10.285	10.1954	10.0979	9.5465	10.1147	14.0888	9.5867
13 (4-10)	2.497	2.2300	2.5803	2.4441	2.1849	1.9397	2.2082
14 (4-11)	5.397	5.1186	5.3769	4.0831	6.4570	5.1783	4.8076
15 (4-17)	16.503	14.0053	16.0581	12.3422	18.2837	20.6111	17.1329
16 (5-6)	8.193	8.9713	8.6789	8.6568	8.2311	7.2776	7.9300
17 (5-11)	3.829	4.0756	3.3199	4.0585	3.1540	7.7874	3.3090
18 (5-12)	6.151	5.9211	6.4966	5.2401	5.8960	6.8604	6.4789
19 (5-18)	10.465	10.6915	10.8804	13.4226	19.8678	13.7167	10.8960
20 (6-7)	13.925	10.6220	14.0056	13.3777	13.5511	14.1815	13.4840
21 (6-12)	4.415	4.5064	5.0843	6.0785	5.5768	8.1916	5.2432
22 (6-13)	6.863	8.4086	6.9952	8.0438	6.7898	13.9162	6.8823
23 (6-19)	1.769	5.8405	1.0270	1.0226	1.0175	4.5669	1.8317
24 (7-13)	4.339	5.0342	4.3788	4.2592	4.1502	3.9236	4.9901
25 (8-9)	2.115	3.8932	2.1951	2.1809	2.7295	2.6394	2.9352
26 (8-14)	4.951	6.1647	4.2562	3.9382	4.1142	1.2307	4.4302
27 (8-15)	4.147	6.8990	4.6605	5.2629	5.8905	11.0933	6.2621
28 (8-21)	6.044	11.6387	8.8694	12.1908	11.5388	58.6796	12.9061
29 (9-10)	3.222	3.8343	3.2333	4.1286	4.3093	3.1521	4.0837
30 (9-15)	1.970	1.4772	1.7611	3.6731	1.8975	2.1452	1.3963
31 (9-16)	4.290	1.3075	3.2831	2.6622	2.5412	2.1267	2.4584
32 (9-22)	8.020	4.4876	7.1936	4.8959	4.6417	1.0710	4.3971
33 (10-11)	4.857	6.0196	4.9840	5.4788	5.4994	3.4763	5.5176
34 (10-16)	3.689	2.6729	3.6672	3.4673	2.4481	1.9475	2.5489
35 (10-17)	2.831	1.6342	2.4062	2.4088	2.0894	1.1114	2.7184
36 (10-23)	1.985	1.8410	2.1576	1.1880	1.7796	1.5194	2.1896
37 (11-12)	6.373	6.8841	7.1043	7.1520	7.9676	10.6004	7.4787
38 (11-17)	4.865	4.1393	5.2070	4.3348	4.9463	3.8586	5.1103

39 (11-18)	3.412	3.3264	3.6853	2.5376	3.3697	4.0934	3.6076
40 (11-24)	1.027	1.0000	1.0007	1.0364	1.0136	1.0000	1.0328
41 (12-13)	6.218	6.9373	6.6302	6.2738	6.9306	12.9063	6.8516
42 (12-18)	7.342	4.4568	6.6773	5.7109	6.4813	4.7955	6.5476
43 (12-19)	5.458	4.6758	5.2167	6.1412	5.3985	4.1583	4.7652
44 (12-25)	1.140	1.0084	1.0016	1.0061	1.0063	1.0000	1.0080
45 (13-19)	7.401	7.5103	8.1289	8.9413	7.8888	15.3458	7.3394
46 (13-20)	4.578	5.2449	4.5151	4.7986	4.3830	4.4623	4.4722
47 (13-26)	1.561	1.0550	1.0010	1.0556	1.0086	1.2463	1.2093
Best weight (kg)	10435.47	10504.20	10326.296	10400.079	10283.094	12673.26	10268.06
Average weight (kg)	10658.48	10590.67	10399.828	10707.902	10379.632	14860.15	10303.99
Worst weight (kg)	-	-	-	11202.677	10491.617	16863.29	10350.43
SD (kg)	129.90	52.51	75.441	250.624	57.586	1336.49	24.77
Max_{NFE}	80000	20000	25000	20000	20000	20000	20000

Table 10: The first five natural frequencies of the 1410-bar dome found by various methods (Hz)

Frequeny numbe	CPA [11]	ECBO-Cascade [9]	PFJA [58]	JA [50]	ST-JA [50]	This study	
						AOA	IAOA
1	7.000	7.0020	7.0009	7.0017	7.0002	7.0050	7.0000
2	7.000	7.0030	7.0009	7.0017	7.0002	7.0050	7.0000
3	9.000	9.0010	9.0001	9.0027	9.0006	9.0021	9.0000
4	9.002	9.0010	9.0002	9.0035	9.0021	9.0021	9.0000
5	9.002	9.0030	9.0002	9.0035	9.0021	9.0254	9.0000

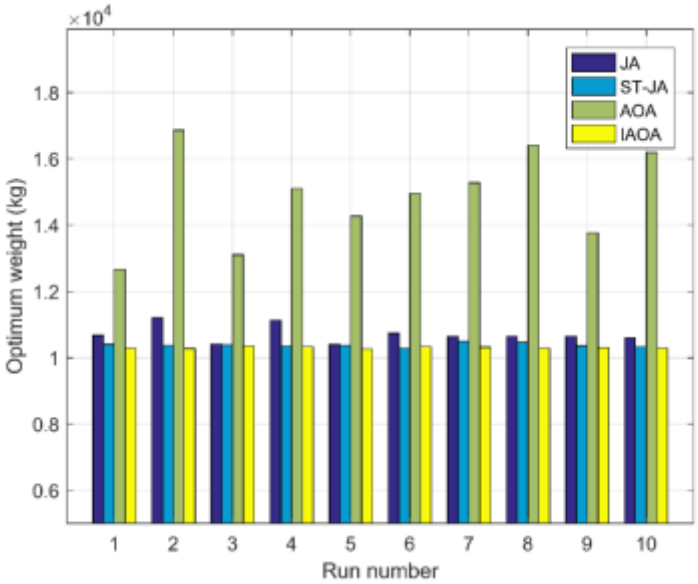

Figure 14. Diversity of the optimal weights of the 1410-bar truss found by different methods

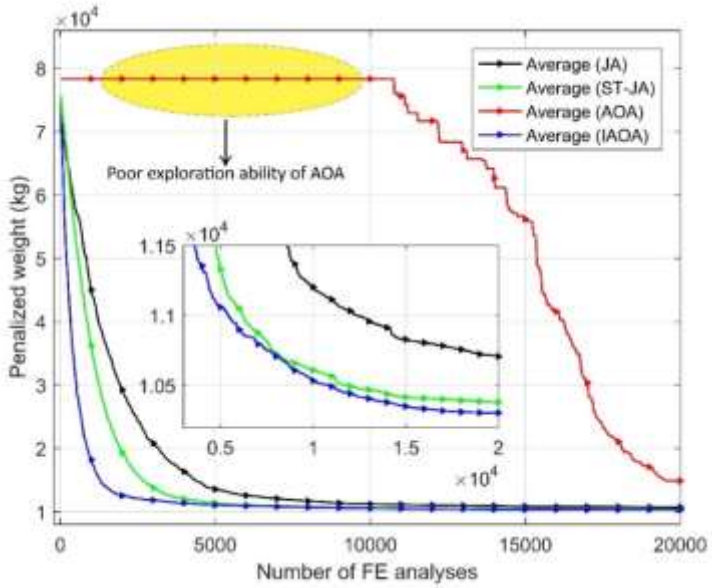

Figure 15. The weight convergence histories of the 1410-bar truss obtained by different methods

4. CONCLUSION

This study has successfully applied an improved variant of the arithmetic optimization algorithm (AOA), called IAOA, to solve truss optimization problems with frequency constraints. The arithmetic optimization algorithm (AOA) is a recently proposed metaheuristic optimization algorithm inspired by the distribution characteristics of the four basic arithmetic operations of addition, subtraction, multiplication, and division, and has been applied to a number of optimization problems. The original AOA, however, due to its poor exploration capability, suffers severely from the problem of slow convergence rate and premature convergence to non-optimal solutions, especially when dealing with high-dimensional optimization problems. More recently, to overcome the drawbacks of the original AOA, the IAOA has been proposed and applied to discrte structural optimization. The major contributions of IAOA include: (1) In the proposed IAOA, both exploration and exploitation phases of the original AOA are modified to improve the convergence speed and the exploration capability; (2) Compared to the original AOA, IAOA requires fewer algorithm-specific parameters, which makes it easier to implement. Finally, to verify the efficiency and effectiveness of the IAOA, three large-scale dome-like truss optimization problems with multiple frequency constraints have been examined. To the best of our knowledge, this is the first attempt to apply both the AOA and IAOA for structural optimization problems with frequency constraints. The results achieved by the IAOA have been compared with those of the original AOA and other methods existing in the literature. The results have demonstrated that the IAOA meaningfully outperforms the original AOA in both terms of convergence speed and robustness. It has been indicated that the IAOA can effectively overcome the shortcomings of the original AOA, such as poor exploration capability and slow convergence rate. In addition, the result comparisons with other methods

in the literature have shown than the IAOA is comparable to or better than many other existing methods. Accordingly, IAOA offers a great potential to extend the applications to other types of structural optimization problems, including frames, plates, shells, etc.

REFERENCES

1. Khatibinia M, Naseralavi SS. Truss optimization on shape and sizing with frequency constraints based on orthogonal multi-gravitational search algorithm, *J Sound Vib* 2014; **333**(24) :6349-69. https://doi.org/10.1016/j.jsv.2014.07.027.
2. Grandhi R, Venkayyat VB. Structural optimization with frequency constraints, *AIAA J* 1988; **26**(7): 858-66. https://doi.org/10.2514/3.9979.
3. Bellagamba L, Yang TY. Minimum-mass truss structures with constraints on fundamental natural frequency, *AIAA J* 1981; **19**(11):1452-8. https://doi.org/10.2514/3.7875.
4. Tong WH, Liu GR. An optimization procedure for truss structures with discrete design variables and dynamic constraints, *Comput Struct* 2001; **79**(2): 155-62. https://doi.org/10.1016/S0045-7949(00)00124-3.
5. Sedaghati R, Suleman A, Tabarrok B. Structural optimization with frequency constraints using the finite element force method, *AIAA J* 2002; **40**(2): 382-8. https://doi.org/10.2514/2.1657.
6. Lingyun W, Mei Z, Guangming W, Guang M. Truss Optimization on Shape and Sizing with Frequency Constraints Based on Genetic Algorithm, *Comput Mech* 2005; **35**(5): 361-8. https://doi.org/10.1007/s00466-004-0623-8.
7. Gomez HM. Truss optimization with dynamic constraints using a particle swarm algorithm, *Expert Syst Appl* 2011; **38**: 957-68. https://doi.org/10.1016/j.eswa.2010.07.086.
8. Kaveh A, Zolghadr A. Truss optimization with natural frequency constraints using a hybridized CSS-BBBC algorithm with trap recognition capability, *Comput Struct* 2012; **102**: 14-27. https://doi.org/10.1016/j.compstruc.2012.03.016.
9. Kaveh A, Ilchi Ghazaan M. Optimal design of dome truss structures with dynamic frequency constraints, *Struct Multidisc Optim* 2016; **53**(3): 605-21. https://doi.org/10.1007/s00158-015-1357-2.
10. Ho-Huu V, Nguyen-Thoi T, Truong-Khac T, Le-Anh L, Vo-Duy T. An improved differential evolution based on roulette wheel selection for shape and size optimization of truss structures with frequency constraints, *Neural Comput Appl* 2018; **29**(1): 167-85. https://doi.org/10.1007/s00521-016-2426-1.
11. Kaveh A, Zolghadr A. Optimal design of cyclically symmetric trusses with frequency constraints using cyclical parthenogenesis algorithm, *Adv Struct Eng* 2018; **21**(5): 739-55. https://doi.org/10.1177%2F1369433217732492.
12. Lieu QX, Do DT, Lee J. An adaptive hybrid evolutionary firefly algorithm for shape and size optimization of truss structures with frequency constraints, *Comput Struct* 2018; **195**: 99-112. https://doi.org/10.1016/j.compstruc.2017.06.016.
13. Kaveh A, Biabani Hamedani K, Kamalinejad M. Set theoretical variants of the teaching–learning-based optimization algorithm for optimal design of truss structures with multiple frequency constraints, *Acta Mech* 2020; **231**(9): 3645-72. https://doi.org/10.1007/s00707-020-02718-3.

14. Grandhi R. Structural optimization with frequency constraints-a review, *AIAA J* 1993; **31**(12): 2296-303. https://doi.org/10.2514/3.11928.
15. Talbi EG. *Metaheuristics: from Design to Implementation*, John Wiley & Sons, 1st edition, USA, 2009.
16. Kaveh A. *Advances in Metaheuristics Algorithms for Optimal Design of Structures*, Springer, 3rd edition, Cham, Switzerland, 2021.
17. Mirjalili S, Lewis A. The whale optimization algorithm, *Adv Eng Softw* 2016; **95**: 51-67. https://doi.org/10.1016/j.advengsoft.2016.01.008.
18. Holland JH. Genetic algorithms, *Sci Am* 1992; **267**(1): 66-73.
19. Dasgupta D, Michalewicz Z, editors. *Evolutionary algorithms in engineering applications*, Springer Science & Business Media, 2013.
20. Nordin P, Keller RE, Francone FD. *Genetic Programming*, Banzhaf W, editor, Springer, 1998.
21. Fogel LJ. *Intelligence through Simulated Evolution: Forty Years of Evolutionary Programming*, John Wiley & Sons, 1999.
22. Moscato P. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms, Technical Report Caltech Concurrent Computation Program, Report 826, California Institute of Technology, Pasadena, CA; 1989.
23. Storn R, Price K. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, *J Glob Optim* 1997; **11**(4): 41-59. https://doi.org/10.1023/A:1008202821328.
24. Rashedi E, Nezamabadi-Pour H, Saryazdi S. GSA: a gravitational search algorithm, *Inf Sci* 2009; **179**(13): 2232-48. https://doi.org/10.1016/j.ins.2009.03.004.
25. Genç HM, Eksin I, Erol OK. Big Bang-Big Crunch optimization algorithm hybridized with local directional moves and application to target motion analysis problem, In *2010 IEEE International Conference on Systems, Man and Cybernetics* 2010, pp. 881-887. IEEE. https://doi.org/10.1109/ICSMC.2010.5641871.
26. Kaveh A, Talatahari S. A novel heuristic optimization method: charged system search, *Acta Mech* 2010; **213**(3): 267-89. https://doi.org/10.1007/s00707-009-0270-4.
27. Kaveh A, Khayatazad M. A new meta-heuristic method: ray optimization, *Comput Struct* 2012; **112**: 283-94. https://doi.org/10.1016/j.compstruc.2012.09.003.
28. Kaveh A, Bakhshpoori T. Water evaporation optimization: a novel physically inspired optimization algorithm, *Comput Struct* 2016; **167**: 69-85. https://doi.org/10.1016/j.compstruc.2016.01.008.
29. Karaboga D. An idea based on honey bee swarm for numerical optimization. Technical report-tr06, Erciyes University, engineering faculty, computer engineering department; 2005.
30. Kennedy J, Eberhart R. Particle swarm optimization, *In Proceedings of ICNN'95-International Conference on Neural Networks* 1995; **4**: 1942-1948. IEEE. https://doi.org/10.1109/ICNN.1995.488968.
31. Dorigo M, Birattari M, Stutzle T. Ant colony optimization, *IEEE Comput Intell Mag* 2006; **1**(4): 28-39. https://doi.org/10.1109/MCI.2006.329691.
32. Yang XS, Deb S. Cuckoo search via Lévy flights. In *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)* 2009, pp. 210-214, IEEE. https://doi.org/10.1109/NABIC.2009.5393690.

33. Geem ZW, Kim JH, Loganathan GV. A new heuristic optimization algorithm: harmony search, *Simulation* 2001; **76**(2): 60-8. https://doi.org/10.1177%2F003754970107600201.
34. Kashan AH. League championship algorithm: a new algorithm for numerical function optimization. In 2009 international conference of soft computing and pattern recognition 2009 Dec 4 (pp. 43-48). IEEE. https://doi.org/10.1109/SoCPaR.2009.21.
35. Rao RV, Savsani VJ, Vakharia DP. Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems, *Comput Aided Des* 2011; **43**(3): 303-15. https://doi.org/10.1016/j.cad.2010.12.015.
36. Atashpaz-Gargari E, Lucas C. Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. *In 2007 IEEE Congress on Evolutionary Computation* 2007, pp. 4661-4667. IEEE. https://doi.org/10.1109/CEC.2007.4425083.
37. Wagan AI, Shaikh MM. A new metaheuristic optimization algorithm inspired by human dynasties with an application to the wind turbine micrositing problem, *Appl Soft Comput* 2020; **90**: 106176. https://doi.org/10.1016/j.asoc.2020.106176.
38. Gandomi AH. Interior search algorithm (ISA): a novel approach for global optimization, *ISA Trans* 2014; **53**(4): 1168-83. https://doi.org/10.1016/j.isatra.2014.03.018.
39. Abualigah L, Diabat A, Mirjalili S, Abd Elaziz M, Gandomi AH. The arithmetic optimization algorithm, *Comput Meth Appl Mech Eng* 2021; **376**: 113609. https://doi.org/10.1016/j.cma.2020.113609.
40. Khatir S, Tiachacht S, Le Thanh C, Ghandourah E, Mirjalili S, Wahab MA. An improved Artificial Neural Network using Arithmetic Optimization Algorithm for damage assessment in FGM composite plates, *Compos Struct* 2021; **273**: 114287. https://doi.org/10.1016/j.compstruct.2021.114287
41. Mirza AF, Mansoor M, Zerbakht K, Javed MY, Zafar MH, Khan NM. High-efficiency hybrid PV-TEG system with intelligent control to harvest maximum energy under various non-static operating conditions, *J Clean Prod* 2021; **320**: 128643. https://doi.org/10.1016/j.jclepro.2021.128643.
42. Xu YP, Tan JW, Zhu DJ, Ouyang P, Taheri B. Model identification of the Proton Exchange Membrane Fuel Cells by Extreme Learning Machine and a developed version of Arithmetic Optimization Algorithm, *Ener Rep* 2021; **7**: 2332-42. https://doi.org/10.1016/j.egyr.2021.04.042.
43. Abualigah L, Diabat A, Sumari P, Gandomi AH. A novel evolutionary arithmetic optimization algorithm for multilevel thresholding segmentation of covid-19 CT images, *Proces* 2021; **9**(7): 1155. https://doi.org/10.3390/pr9071155.
44. Kaveh A, Biabani Hamedani K. Improved arithmetic optimization algorithm and its application to discrete structural optimization, *Structures* 2021 (accepted for publication).
45. Kaveh A, Biabani Hamedani K, Barzinpour F. Optimal size and geometry design of truss structures utilizing seven meta-heuristic algorithms: a comparative study, *Int J Optim Civil Eng* 2020; **10**(2): 231-60.
46. Kaveh A, Kamalinejad M, Biabani Hamedani K. Enhanced versions of the shuffled shepherd optimization algorithm for the optimal design of skeletal structures, *Structures* 2021; **29**: 1463-95. https://doi.org/10.1016/j.istruc.2020.12.032.
47. Kaveh A, Biabani Hamedani K. Set theoretical variants of optimization algorithms for optimal design of skeletal structures, *Int J Optim Civil Eng* 2020; **10**(4): 669-700.

48. Kaveh A, Zolghadr A. Democratic PSO for truss layout and size optimization with frequency constraints, *Comput Struct* 2014; **130**: 10-21. https://doi.org/10.1016/j.compstruc.2013.09.002.
49. Kaveh A, Kamalinejad M, Biabani Hamedani K, Arzani H. Quantum Teaching-Learning-Based Optimization algorithm for sizing optimization of skeletal structures with discrete variables, *Struct* 2021; **32**:1798-819. https://doi.org/10.1016/j.istruc.2021.03.046.
50. Kaveh A, Biabani Hamedani K, Joudaki A, Kamalinejad M. Optimal analysis for optimal design of cyclic symmetric structures subject to frequency constraints, *Struct* 2021; **33**: 3122-36. https://doi.org/10.1016/j.istruc.2021.06.054.
51. Kaveh A, Biabani Hamedani K, Kamalinejad M. An enhanced Forensic-Based Investigation algorithm and its application to optimal design of frequency-constrained dome structures, *Comput Struct* 2021; **256**: 106643. https://doi.org/10.1016/j.compstruc.2021.106643.
52. Goldfarb L. Cognitive interferences and their development in the context of numerical tasks: review and implications. In: Henik A, Fias W, editors. Heterogeneity of function in numerical cognition: Academic Press; 2018, pp. 245-262. https://doi.org/10.1016/C2016-0-00729-5.
53. Liu J, Wu C, Wu G, Wang X. A novel differential search algorithm and applications for structure design, *Appl Math Comput* 2015; **268**: 246-69. https://doi.org/10.1016/j.amc.2015.06.036.
54. Kaveh A, Biabani Hamedani K, Kamalinejad M, Joudaki A. Quantum-based jellyfish search optimizer for structural optimization, *Int J Optim Civil Eng* 2021; **11**(2): 329-56.
55. Kaveh A, Ilchi Ghazaan M. Vibrating particles system algorithm for truss optimization with multiple natural frequency constraints, *Acta Mech* 2017; **228**(1): 307-22. https://doi.org/10.1007/s00707-016-1725-z.
56. Carvalho JP, Lemonge AC, Carvalho EC, Hallak PH, Bernardino HS. Truss optimization with multiple frequency constraints and automatic member grouping, *Struct Multidisc Optim* 2018; **57**(2): 547-77. https://doi.org/10.1007/s00158-017-1761-x.
57. Grzywinski M, Dede T, Ozdemir YI. Optimization of the braced dome structures by using Jaya algorithm with frequency constraints, *Steel Compos Struct* 2019; **30**(1): 47-55. https://doi.org/10.12989/scs.2019.30.1.047/
58. Degertekin SO, Bayar GY, Lamberti L. Parameter free Jaya algorithm for truss sizing-layout optimization under natural frequency constraints, *Comput Struct* 2021; **245**:106461. https://doi.org/10.1016/j.compstruc.2020.106461.
59. Dede T, Grzywinski M, Selejdak J. Continuous size optimization of large-scale dome structures with dynamic constraints, *Struct Eng Mech* 2020; **73**(4): 397-405. https://doi.org/10.12989/sem.2020.73.4.397.
60. Kaveh A, Amirsoleimani P, Eslamlou AD, Rahmani P. Frequency-constrained optimization of large-scale dome-shaped trusses using chaotic water strider algorithm, *Structures* 2021; **32**: 1604-18. https://doi.org/10.1016/j.istruc.2021.03.033.