Journal of Science and Technology in Civil Engineering, HUCE (NUCE), 2022, 16 (2): 22–37

A HYBRID ARITHMETIC OPTIMIZATION ALGORITHM AND DIFFERENTIAL EVOLUTION FOR OPTIMIZATION OF TRUSS STRUCTURES SUBJECTED TO FREQUENCY CONSTRAINTS

Dieu T. T. Do[a], Tan-Tien Nguyen[b], Quoc-Hung Nguyen[b], Tinh Quoc Bui[c,*]

[a]*Duy Tan Research Institute for Computational Engineering, Duy Tan University, 254 Nguyen Van Linh street, Da Nang, Vietnam*
[b]*Faculty of Engineering, Vietnamese-German University, Le Lai street, Thu Dau Mot city, Binh Duong province, Vietnam*
[c]*Department of Civil and Environmental Engineering, Tokyo Institute of Technology, Tokyo, Japan*

Article history:
Received 16/10/2021, Revised 31/01/2022, Accepted 15/02/2022

Abstract

A new hybrid arithmetic optimization algorithm (AOA) associated with differential evolution (DE) is developed for truss optimization. The development is named as ADE with the goal of maintaining a balance between low computational cost and good solution quality. Besides, several limitations of the AOA, which include the inefficiency of the exploration phase and the inconvenient use of two parameters MOA and MOP to find the optimal solution, as well as how to overcome them are also discussed. In terms of AOA in ADE, the exploration phase is removed, and both math optimizer accelerated (MOA) and math optimizer probability (MOP) parameters are adjusted to be independent of the maximum number of iterations. Moreover, the exploitation phase is modified to exploration which helps to limit local solutions and maintain a balance between exploitation and exploration in ADE algorithm. Through a probability parameter, the DE with DE/best/1 operator is executed in ADE to improve exploitation capability as well as convergence rate. Four truss structures with continuous design variables are considered to demonstrate the performance of the current algorithm. The obtained results show that the developed algorithm has a low computational cost, indicating its computational efficiency.

Keywords: arithmetic optimization algorithm; differential evolution; meta-heuristic; truss structure; frequency constraints.

https://doi.org/10.31814/stce.huce(nuce)2022-16(2)-03

1. Introduction

Structural optimization is the process of designing structures under certain constraints to achieve better performance and lower manufacturing costs. Many different types of structures have been investigated in real applications. In particular, truss optimization has been extensively studied as a benchmark problem in a variety of publications using different optimization techniques [1–8]. For example, Kaveh et al. [5] reviewed meta-heuristic methods such as genetic algorithm (GA), particle swarm optimization (PSO), harmony search (HS), firefly algorithm (FA) and several algorithms for structural

*Corresponding author. *E-mail address:* tinh.buiquoc@gmail.com (Bui, T. Q.)

optimization with frequency constraints. Lieu et al. [6] proposed an adaptive hybrid evolutionary firefly algorithm (AHEFA), which is a hybridization of the differential evolution (DE) algorithm and the firefly algorithm (FA) for truss optimization problems. Ho-Huu et al. [9] proposed an improved differential evolution (IDE) method for solving size and shape optimization problems of truss structures. A new selection scheme based on multi-mutation operators was proposed in the IDE's mutation phase to help to maintain an effective balance between exploration and exploitation abilities. When compared to other algorithms in the literature, improvements in IDE help to save computational cost while providing acceptable optimal solutions. Besides, an enhanced differential evolution named as ANDE was proposed by Pham [8] for solving those truss problems. In which, the traditional differential evolution (DE) has been modified in three ways: the adaptive p-best strategy, the directional mutation rule, and the nearest neighbor comparison method. ANDE with these modifications is able to maintain the balance between exploration and exploitation, and help to save the computational cost. Those methods can be divided into two major categories: gradient-based and non-gradient-based algorithms. Optimality criterion (OC) [10], force method [11] and sequential quadratic programming (SQP) [12] , for example, are some of the most common approaches in the first group. Although these approaches have a relatively fast convergence rate, sensitivity analyses are always required. Their mathematical analysis performances are quite complicated and more importantly, they are costly and even unsuccessful in many other cases. Furthermore, the search ability focuses only on derivative data provided by sensitivity analyses; therefore, obtained solutions are frequently trapped in local areas. The non-gradient-based approaches in the second group, also known as metaheuristic methods such as genetic algorithm (GA) [13], differential evolution (DE) [14], flower pollination algorithm (FPA) [15], and their variants [16–18], have been developed to overcome the aforementioned limitations. Sensitivity analysis is no longer required due to stochastic searching techniques that are used to select candidates in a given domain at random. A global optimal solution can be found without a great deal of mathematical expertise. Nonetheless, because of low convergence rate, the process thus takes more effort.

Among the aforementioned algorithms in the second group, arithmetic optimization algorithm (AOA) [19] was proposed recently and has attracted many researchers. The four primary arithmetic operators in mathematics, such as division (D), multiplication (M), addition (A), and subtraction (S), are all used in AOA. AOA is a mathematically implemented and modeled optimization algorithm that works in a vast scope of search spaces. The exploration and exploitation phases are the two main phases of the AOA. In the study [19], although the AOA has applied successfully to solve 29 benchmark functions and 5 real-world engineering problems, it still has several limitations when solving other real-world problems. Consequently, a number of improved AOA versions have been proposed. For example, Agushaka et al. [20] presented an advanced arithmetic optimization algorithm to solve mechanical engineering problems, in which, the optimization process begins by using the beta distribution to initialize the candidate solutions. Moreover, the exponential (E 'e') and natural log operator (L 'ln') are used instead of division (D), multiplication (M) in the exploration. The effectiveness of the method was demonstrated through benchmark functions and three engineering problems. Besides, an improved AOA was proposed to gain an optimal design for a cruise control system in an automobile, in which, the exploration task was handled by AOA, and the exploitation task was handled by another algorithm, the Nelder-Mead. Several other improved versions of AOA can be found as [21, 22].

Besides, DE is a popular non-gradient-based method inspired by nature. Because of its effectiveness in finding a global optimal solution in given spaces, this method has been widely applied to a variety of disciplines [23–26]. Different improved versions of the DE algorithm have been developed to reduce the computational cost or improve the quality of the solutions such as [27, 28]. For example, Huynh et al. [29] proposed Q-learning differential evolution for truss optimization to maintain a balance between exploration and exploitation. Tan and Li [30] introduced a modified version of the DE with mixed mutation strategy based on deep Q-network. According to the theory of no free lunch [31], even though many optimization algorithms have been proposed, none of them can solve all optimization problems. This motivates us to propose a hybrid arithmetic optimization algorithm and differential evolution as called ADE in this work.

This paper aims to apply the developed ADE to optimize truss structures subjected to frequency constraints. In addition, limitations of the AOA are also discussed, as well as how to overcome them. In each iteration, a randomly generated probability parameter is used to determine whether AOA or DE that would be used to generate new candidate solutions in the population. The exploration phase with division (D) and multiplication (M) operators is removed from AOA of ADE because it does not contribute significantly to finding optimal solutions as investigated in the numerical examples. The MOA parameter, which is used to determine whether the exploration or exploitation phase will be carried out, will be modified. MOP parameter is also modified to be independent of the number of iterations and the maximum iteration. Furthermore, in the new algorithm, the exploitation phase is modified to exploration which helps to limit local solutions. The proposed algorithm uses a DE with DE/best/1 operator to improve the exploitation ability as well as convergence rate of the algorithm. Testing for optimization of truss structures with frequency constraints demonstrates the effectiveness of ADE. The optimal results of the proposed method are compared to those obtained by others in the literature.

2. Truss optimization problem

The goal of truss structure optimization problems with frequency constraints is to minimize the weight of the truss by designing member sizes or/and shape. Member cross-sectional areas as well as nodal coordinates have been considered as continuous design variables. Connectivity data of the structure is predetermined and assumed to remain constant throughout the optimization process. Furthermore, each variable is created within a predetermined range. As a result, this issue can be expressed mathematically as

$$\begin{aligned} &\text{Minimize: } f(\mathbf{A},\mathbf{x}) = \sum_{i=1}^{m} \rho_i A_i L_i \left(x_j\right) \\ &\text{Subject to } \begin{cases} \omega_l \geq \omega_l^* \\ \omega_k \leq \omega_k^* \\ A_i^{\min} \leq A_i \leq A_i^{\max} \\ x_j^{\min} \leq x_j \leq x_j^{\max} \end{cases} \end{aligned} \tag{1}$$

where $\mathbf{A} = \{A_1, ..., A_m\}$ and $\mathbf{x} = \{x_1, ..., x_n\}$ are the cross-sectional area and nodal coordinates design variable vectors, respectively; n represents the total number of constraints on nodal coordinates; m represents the total number of members in the structure; the length and the material density of ith member, respectively, are represented by L_i and ρ_i ; the lth and kth natural frequencies of the structure are denoted by ω_l and ω_k , respectively; ω_l^* and ω_k^* symbolize the lower and upper bounds; A_i's lower and upper bounds are $A_i^{\min}$ and $A_i^{\max}$, respectively, while x_j's lower and upper bounds are $x_j^{\min}$ and $x_j^{\max}$, respectively.

The penalty function method, which is one of the most widely used constraint handling approaches [32], is used in this study to convert the constrained optimization problem in Eq. (1) into an unconstrained one. As a result, the above problem can be reformulated as follows:

$$\begin{aligned} f_{\text{cost}}(\mathbf{A},\mathbf{x}) &= (1+\varepsilon_1\upsilon)^{\varepsilon_2} f(\mathbf{A},\mathbf{x}) \\ \upsilon &= \sum_{r=1}^{p} \max\{0, g_r(\mathbf{A},\mathbf{x})\} \end{aligned} \tag{2}$$

In which, υ symbolizes the sum of design constraint violations; $g_r(\mathbf{A},\mathbf{x})$ represents the rth constraint; p represents the number of constraints; the parameters ε_1 and ε_2 are chosen based on the exploration and exploitation rates of the search space. In this study, ε_1 and ε_2 are respectively set to be 1 and 1.5 at the beginning of the iteration and gradually increased by 0.05 in each iteration until it reaches 3 as studied in [6].

3. A hybrid arithmetic optimization algorithm and differential evolution

3.1. Arithmetic optimization algorithm

AOA is inspired by traditional arithmetic operators such as division, multiplication, subtraction, and addition, which are commonly used to study numbers. AOA consists of initialization, exploration and exploitation phases. The AOA's main procedure is briefly described as follows:

- Initialization phase: An initial population of NP individuals is generated at random in a given search space, as follows:

$$x_{i,j} = x_j^{\min} + rand(0,1)\left(x_j^{\max} - x_j^{\min}\right) \tag{3}$$

where $i = 1, 2, ..., NP$; $j = 1, 2, ..., D$; D is the number of design variables; $x_j^{\max}$ and $x_j^{\min}$ are the upper and lower bounds of the $x_{i,j}$; $rand(0,1)$ is a random number with a uniform distribution within the range [0, 1].

The Math Optimizer Accelerated (MOA) function, which is used to select exploration or exploitation phases, is calculated as follows:

$$MOA(cIter) = Min + cIter \times \left(\frac{Max - Min}{mIter}\right) \tag{4}$$

where $cIter$ and $mIter$ symbolize the current iteration and maximum number of iterations, respectively; the terms Min and Max represent the accelerated minimum and maximum values of the function, respectively.

- Exploration phase: In this phase, if a random number $r_1 > MOA$, new candidates are generated by using the Division (D), or Multiplication (M) operators, which aims to reinforce exploration ability, as described below:

$$x_{i,j} = \begin{cases} x_j^{best} \div (MOP + \varepsilon) \times \left(\left(UB_j - LB_j\right) \times \mu + LB_j\right), & r_2 < 0.5 \\ x_j^{best} \times MOP \times \left(\left(UB_j - LB_j\right) \times \mu + LB_j\right), & \text{otherwise} \end{cases} \tag{5}$$

in which x_j^{best} is the jth position in the best solution obtained so far; ε is a small number; the lower and upper bound values of the jth position are denoted by LB_j and UB_j, respectively; μ is a control parameter for adjusting the search process, and it is set to be 0.5; r_2 is a random number in the range [0, 1]; Math Optimizer probability called MOP is a coefficient and defined as follows

$$MOP\,(cIter) = 1 - \frac{cIter^{1/\alpha}}{mIter^{1/\alpha}} \tag{6}$$

where the value of α is set to be 5.

- Exploitation phase: If $r_1 \leq MOA$, either subtraction or addition operators is performed to find the near-optimal solutions that may be discovered after several iterations. This search strategy is described as follows:

$$x_{i,j} = \begin{cases} x_j^{best} - MOP \times \left(\left(UB_j - LB_j\right) \times \mu + LB_j\right), & r_3 < 0.5 \\ x_j^{best} + MOP \times \left(\left(UB_j - LB_j\right) \times \mu + LB_j\right), & otherwise \end{cases} \tag{7}$$

in which r_3 is a random number in the range [0, 1].

3.2. Differential evolution

The differential evolution (DE) is a population-based algorithm that was first introduced by Storn and Price [14]. Four major phases of DE are as follows:

- Initialization phase: Eq. (3) is used to generate individuals in the initial population, just as it is in the initialization phase of AOA.

- Mutation phase: Then, using mutation operations, each individual $\mathbf{x}_i$ in the population is used to create a mutant vector $\mathbf{v}_i$. The DE frequently employs the following mutation operations:

$$\begin{aligned} &\text{DE/rand/1: } \mathbf{v}_i = \mathbf{x}_{R_1} + F \times (\mathbf{x}_{R_2} - \mathbf{x}_{R_3}) \\ &\text{DE/best/1: } \mathbf{v}_i = \mathbf{x}_{best} + F \times (\mathbf{x}_{R_1} - \mathbf{x}_{R_2}) \\ &\text{DE/rand/2: } \mathbf{v}_i = \mathbf{x}_{R_1} + F \times (\mathbf{x}_{R_2} - \mathbf{x}_{R_3}) + F \times (\mathbf{x}_{R_4} - \mathbf{x}_{R_5}) \\ &\text{DE/best/2: } \mathbf{v}_i = \mathbf{x}_{best} + F \times (\mathbf{x}_{R_1} - \mathbf{x}_{R_2}) + F \times (\mathbf{x}_{R_3} - \mathbf{x}_{R_4}) \end{aligned} \tag{8}$$

where R_1, R_2, R_3, R_4, R_5 are integers chosen at random from $1, 2, ..., NP$ and must satisfy $R_1 \neq R_2 \neq R_3 \neq R_4 \neq R_5 \neq i$; F is the scale factor selected at random from [0, 1]; $\mathbf{x}_{best}$ is the best individual in the current population.

- Crossover phase: Following the completion of mutation, each target vector $\mathbf{x}_i$ creates a trial vector $\mathbf{u}_i$ by binomial crossovering several elements of the vector $\mathbf{x}_i$ with elements of the mutant vector $\mathbf{v}_i$.

$$u_{i,j} = \begin{cases} v_{i,j} \text{ if } rand\,[0,1] \leq C_r \text{ or } j = j_{rand} \\ x_{i,j} \text{ otherwise} \end{cases} \tag{9}$$

where $i = 1, 2, ..., NP$; $j = 1, 2, ..., D$; the integer j_{rand} is chosen from 1 to D, and the crossover control parameter C_r is chosen from the range [0, 1].

- Selection phase: Finally, the target vector $\mathbf{x}_i$ is compared to each trial vector $\mathbf{u}_i$. The one that is better value will be passed down to the next generation.

$$\mathbf{x}_i = \begin{cases} \mathbf{u}_i \text{ if } f\,(\mathbf{u}_i) \leq f\,(\mathbf{x}_i) \\ \mathbf{x}_i \text{ otherwise} \end{cases} \tag{10}$$

3.3. A hybrid arithmetic optimization algorithm and differential evolution

ADE is a hybrid algorithm that combines AOA and DE to reduce the computational cost which has been shown in this section. ADE includes three major phases described as follows:

- Initialization phase: An initial population with NP individuals is created randomly as in the initialization phase of AOA or DE.

- Exploration phase with modified Arithmetic Optimization Algorithm: Firstly, MOP and MOA in the AOA are adjusted to be independent of the number of iterations and the maximum number of iterations. Because it will be more convenient to solve complex problems without having to limit the number of iterations. MOA is updated as follows:

+ From the formulations of MOA and MOP, it can be seen that MOA starts with a small value (nearly 0) and gradually increases after each iteration, eventually reaching a greater value (nearly 1) in the final iteration, whereas MOP does the opposite. This allows exploration to be employed at an early stage of the search process and exploitation to be done later. Therefore, after investigating, MOA and MOP are set to be 0.4 and 0.7 in the first iteration, respectively. These values will help to improve the convergence rate.

+ If the solution obtained by the current MOA at ith iteration is better than $\mathbf{x}_i$ then both MOA and MOP are kept. Because the MOA and MOP parameter values provide useful information for the search for the optimal solution.

+ Otherwise, $MOA = MOA + \beta$ and $MOP = MOP - \beta$. In which β is a small value. In this study, β is set to be 10^{-3}. These formulas help with the transition from exploration to exploitation. If $MOA > 0.9$, MOA is created randomly in the range [0.4, 0.9]. Besides, if $MOP < 0.2$, MOP is created randomly in the range [0.2, 0.7]. The ranges of values of two parameters, MOA and MOP, have been investigated by the authors and selected appropriate values for the problems in this study. For the sake of brevity, the authors will not present this survey in the study.

The goal of this update is to improve exploration ability in the early stage while also increasing exploitation ability later on. It aids in the search process by lowering computational cost and limiting local solutions.

Secondly, according to our survey, implementing exploration phase does not actually improve the quality of solutions, so it is recommended that the exploration phase should be removed from ADE algorithm. It is demonstrated in the numerical example part.

Next, exploitation phase of the AOA is performed; however, x_j^{best} in Eq. (7) is replaced by x_j^k which is chosen randomly in the population. This helps to improve exploration ability in this phase.

From the above modifications, it can be seen that exploration ability is reinforced in this phase.

- Exploitation phase with DE: Das et al. [33] found that the balance of exploitation and exploration abilities has a significant impact on the success of most population-based optimization algorithms. In which the exploration ability refers to the global search capability, which has a significant impact on the accuracy of the achieved optimal solution. The exploitation describes the ability to perform local searches, which has a significant impact on the convergence of the algorithm. Clearly, if the exploration ability is greater than the other, a global optimal solution can be found, but convergence is slow. This is because the algorithm must require a significant amount of computational cost in order to find the best solution in a given domain. The algorithm, on the other hand, converges quickly, but local optimum solutions may emerge. As a result, if the above two abilities are adjusted to achieve a better balance, the solution accuracy and convergence rate can be achieved at the same time. From above discussion, it can be seen that the exploitation should be reinforced in this phase. As a result, DE algorithm with DE/best/1 operator is used to balance between the exploration and exploitation abilities in ADE. The flowchart of the proposed ADE algorithm is depicted in Fig. 1.

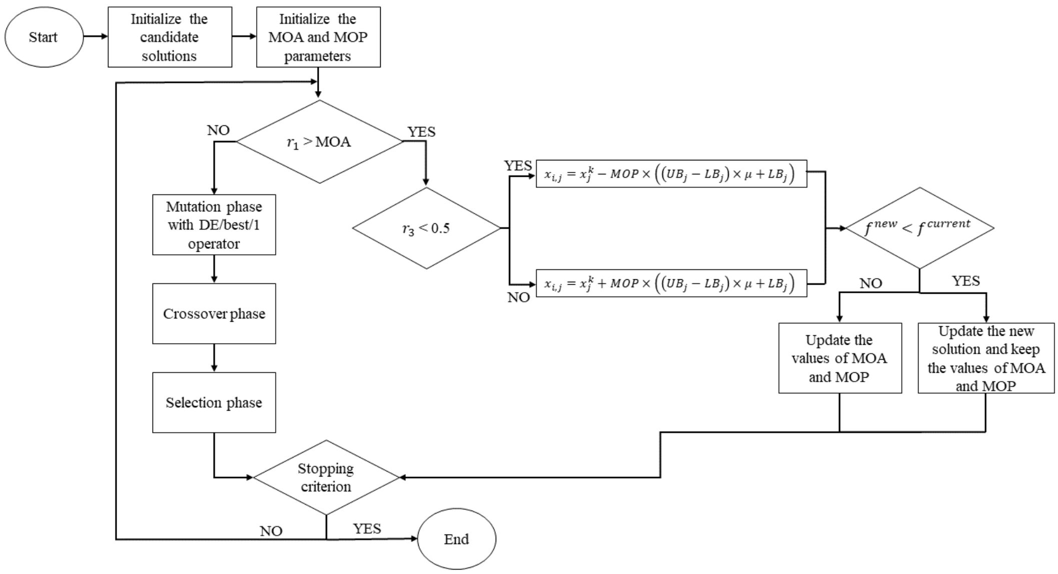

Figure 1. Flowchart of the proposed ADE algorithm

4. Numerical examples

Four truss optimization problems with frequency constraints are investigated to show the efficiency of ADE in terms of the computational cost and quality of the solution. The original AOA, DE, and several other algorithms are used as reference solutions for our comparison purpose. In which, DE with DE/rand/1 operator is used for comparison. Similar to previous studies, a population size NP of 20 is used in all examples. F and C_r are set to be 0.8 and 0.9, respectively for all examples. The values of F and C_r are the same as the exploitation phase with DE of ADE algorithm. The truss analysis is performed with a two-node linear bar element. The optimization process is terminated when the relative error between the best and mean objective function values of the population is less than or equal to the specified tolerance, or when the maximum number of structural analyses (*MaxEval*) is reached. In this study, tolerance is set to be 10^{-6} for all problems. Each of the algorithms is run 30 independent times as same as the previous examples. *MaxEval* is set to be 20000 for the 10-bar truss problem and to 40000 for the others. Data for truss problems is tabulated in Table 1.

Table 1. Data for four truss structures

Problem	Young's modulus E (N/m^2)	Material density ρ (kg/m^3)	Added mass (kg)	Frequency constraints (Hz)
10-bar planar truss	6.98×10^{10}	2770	454	$\omega_1 \geq 7, \omega_2 \geq 15, \omega_3 \geq 20$
72-bar space truss	6.98×10^{10}	2770	2270	$\omega_1 = 4, \omega_3 \geq 6$
200-bar planar truss	2.1×10^{11}	7860	100	$\omega_1 \geq 5, \omega_2 \geq 10, \omega_3 \geq 15$
52-bar dome truss	2.1×10^{11}	7800	50	$\omega_1 \leq 15.9155, \omega_2 \geq 28.6479$

4.1. 10-bar planar truss

The first example deals with a planar truss comprised of ten bars as shown in Fig. 2. The cross-sectional areas of 10 bars are considered as 10 continuous design variables with the boundary condition $0.645 \times 10^{-4} \leq A \leq 50 \times 10^{-4}$. A non-structural mass is added to all free nodes of the structure as shown in the same figure.

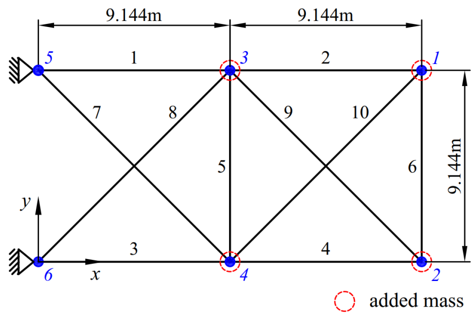

Figure 2. The 10-bar planar truss

A comparison on the numerical results among the developed method and other algorithms is presented in Table 2, in which the effectiveness of the exploitation and exploration phases of AOA is also investigated. From the table, it can be seen that ADE requires fewer finite element analyses than DE and AOA methods to get the optimal solution. Despite the fact that ADE performs more evaluations than IDE and ANDE (6960 analyses for ADE, 6260 analyses for IDE and 6115 analyses for ANDE), the best solution obtained by ADE is superior to those two methods. Obviously, the present method requires the least number of FE analyses to reach an optimal solution whilst guaranteeing the quality of the solution. Moreover, AOA with only the exploration phase is ineffective and even violates constraints; therefore, it is removed from the algorithm. Furthermore, natural frequencies gained by the present method satisfies all frequency constraints as summarized in Table 3. From the above discussions, it can be found that ADE has the ability to strike a balance between computational cost and quality of solution.

Table 2. Optimized designs for 10-bar truss structure gained by the algorithms

Design variables A_i (cm^2)	DE	PSO [34]	HS [35]	IDE [9]	ANDE [8]	AOA	AOA with only exploration phase	AOA with only exploitation phase	ADE
1	35.1056	37.712	34.282	35.0606	35.1829	35.2879	32.3598	36.3890	35.1932
2	14.7244	9.959	15.653	14.6851	14.5442	14.6805	17.6400	14.9800	14.6976
3	35.1445	40.265	37.641	35.0687	35.3286	34.2632	43.8873	34.9081	35.0309
4	14.6804	16.788	16.058	14.8095	14.6738	15.0572	19.4631	14.8197	14.7868
5	0.6450	11.576	1.069	0.6451	0.6450	0.6450	29.6481	0.6450	0.6451
6	4.5604	3.955	4.740	4.5578	4.5703	4.5699	6.7082	4.5474	4.5570
7	23.7704	25.308	22.505	23.5271	23.6857	23.8956	11.4955	23.7094	23.5778
8	23.6519	21.613	24.603	23.7998	23.9418	23.6186	25.8477	23.5835	23.7686
9	12.3541	11.576	12.867	12.5038	12.2272	12.2494	14.7538	11.8976	12.4797
10	12.4878	11.186	12.099	12.4599	12.3616	13.0026	25.0105	12.1935	12.4034
Best weight (kg)	524.453	537.98	529.09	524.4627	524.4956	525.3479	664.6592	524.9197	524.4556
No. FE analysis	17600	-	-	6260	6115	20000	20000	20000	6960
Worst weight (kg)	530.6943	-	-	530.8448	534.3302	798.0008	896.632	534.3918	531.6511
Average weight (kg)	525.4986	540.89	-	525.6162	525.3544	569.1844	758.8502	528.6061	527.4052
Standard deviation	2.3423	6.84	-	2.3041	1.9951	84.426	55.1496	3.0588	3.0507

Table 3. The first eight optimal frequencies of the 10-bar truss gained by the algorithms

Frequency number	DE	PSO [34]	HS [35]	IDE [9]	ANDE [8]	AOA	AOA with only exploration phase	AOA with only exploitation phase	ADE
1	7.0000	7.000	7.0028	7.0000	7.0000	7.0000	6.9348	7.0000	7.0000
2	16.1903	17.786	16.7429	16.1853	16.2015	16.1782	19.0066	16.2494	16.1899
3	20.0000	20.000	20.0548	20.0000	20.0000	20.0008	21.1672	20.0002	20.0000
4	20.0001	20.063	20.3351	20.0006	20.0052	20.0820	26.9734	20.0388	20.0004
5	28.5562	27.776	28.5232	28.5775	28.5233	28.5627	34.3040	28.3432	28.5609
6	28.9690	30.939	29.2911	-	-	29.2393	49.2161	28.7712	28.9896
7	48.5700	47.297	49.0342	-	-	48.5526	50.2659	48.8067	48.5829
8	51.0656	52.286	54.7451	-	-	51.1230	56.0438	51.2604	51.0885

Convergence histories of the different algorithms in terms of the number of FE analyses are simultaneously depicted in Fig. 3. The figure shows that ADE converges faster than the others while AOA with only exploration phase completely fails to find the optimal solution.

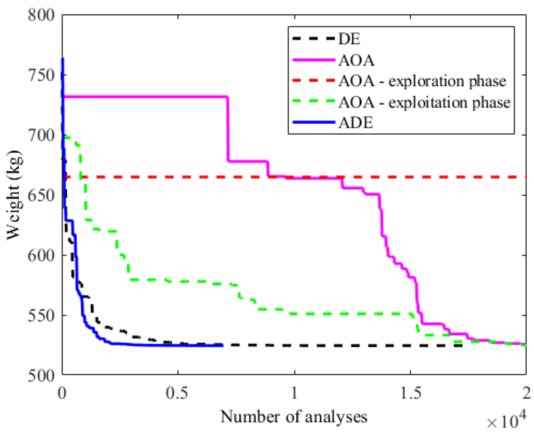

Figure 3. The weight convergence histories of the 10-bar truss

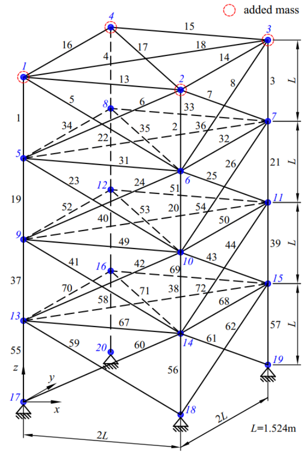

Figure 4. The 72-bar space truss

4.2. *72-bar space truss*

The optimization of the 72-bar truss structure as displayed in Fig. 4 is carried out. Each of the four top nodes of this structure is added a non-structural mass of 2270 kg. Cross-sectional areas of all truss members are divided into 16 groups which correspond to 16 design variables as presented in the first column of Table 4. The boundary condition is $0.645 \times 10^{-4} \leq A \leq 50 \times 10^{-4}$.

A comparison between optimal results achieved by ADE and the other algorithms in the literature is tabulated in Table 4. ADE offers optimal solution better than the other considered approaches. The present method requires only 11400 analyses to reach the optimal solution whereas DE, HS, IDE and AOA require 24640, 50000, 11620 and 40000 analyses, respectively. Although ANDE requires fewer evaluations than the proposed method, the solution obtained by the proposed method is better than that obtained by ANDE. With a standard deviation of 0.0596, ADE is fairly stable. Table 5 details the first five optimal frequencies gained by the various algorithms. None of the violated frequency constraints obtained by ADE is found.

Table 4. Optimized designs for 72-bar truss structure gained by the algorithms

Design variables A_i (cm^2)	DE	PSO [34]	HS [35]	IDE [9]	ANDE [8]	AOA	ADE
1-4	3.4596	2.987	3.6803	3.5863	3.4754	3.3714	3.4111
5-12	7.8528	7.849	7.6808	7.8278	7.8483	7.9593	7.8737
13-16	0.6450	0.645	0.6450	0.6450	0.6450	0.6450	0.6450
17-18	0.6450	0.645	0.6450	0.6450	0.6450	0.6450	0.6450
19-22	7.9739	8.765	9.4955	8.1052	8.0134	9.3141	7.9349
23-30	7.9194	8.153	8.2870	7.8788	7.9316	7.7193	7.9556
31-34	0.6451	0.645	0.6450	0.6451	0.6450	0.6460	0.6450
35-36	0.6451	0.645	0.6461	0.6450	0.6450	0.6450	0.6451
37-40	12.7297	13.450	11.4510	12.5157	12.6420	12.5836	12.7286
41-48	7.9625	8.073	7.8990	8.0102	7.9794	7.9411	7.9331
49-52	0.6450	0.645	0.6473	0.6450	0.6450	0.6450	0.6450
53-54	0.6451	0.645	0.6450	0.6452	0.6450	0.6471	0.6450
55-58	17.0386	16.684	17.4060	16.9997	17.0706	16.1596	17.1160
59-66	8.0168	8.159	8.2736	8.0362	7.9922	8.1451	7.9890
67-70	0.6450	0.645	0.6450	0.6451	0.6450	0.6450	0.6450
71-72	0.6450	0.645	0.6450	0.6453	0.6452	0.6450	0.6451
Best weight (kg)	324.2232	328.823	328.334	324.2441	324.2226	324.7137	324.2028
No. FE analysis	24640	-	50000	11620	8030	40000	11400
Worst weight (kg)	324.3068	-	-	324.6444	324.4292	706.7702	324.4245
Average weight (kg)	324.2440	-	332.640	324.3379	324.2620	380.0288	324.2757
Standard deviation	0.0196	-	2.390	0.1023	0.04760	130.4063	0.0596

Table 5. The first five optimal frequencies of the 72-bar truss gained by the algorithms

Frequency number	DE	PSO [34]	HS [35]	IDE [9]	ANDE [8]	AOA	ADE
1	4.000	4.000	4.000	4.000	4.000	4.000	4.000
2	4.000	4.000	4.000	4.000	4.000	4.000	4.000
3	6.000	6.000	6.000	6.000	6.000	6.000	6.000
4	6.268	6.219	6.2723	6.278	6.2698	6.293	6.264
5	9.099	8.976	9.0749	9.112	9.1012	9.091	9.095

4.3. 200-bar planar truss

Next study is devoted to numerical investigation of a 200-bar planar truss structure as shown in Fig. 5. Non-structural mass of 100 kg is added to each of the upper nodes of this truss structure as indicated in the same figure. The structure consists of 29 member groups that are considered as 29 design variables. The boundary condition of this structure is $1 \times 10^{-5} \leq A \leq 25 \times 10^{-4}$.

In this example, the optimal solutions gained by ADE and DE show nearly the same between each other, and they are better than those of CSS-BBBC as indicated in Table 6. The number of FE analyses required by ADE to find the optimal solution is less than that required by DE, CSS-BBBC, and AOA. In this case, ANDE obtains the best solution with the fewest number of evaluations, but it is unstable with a standard deviation of 33.4775, whereas the proposed algorithm ADE and DE maintain stability with a standard deviation of less than 0.1. Moreover, the natural frequencies achieved by ADE do not show any violation as shown in Table 7.

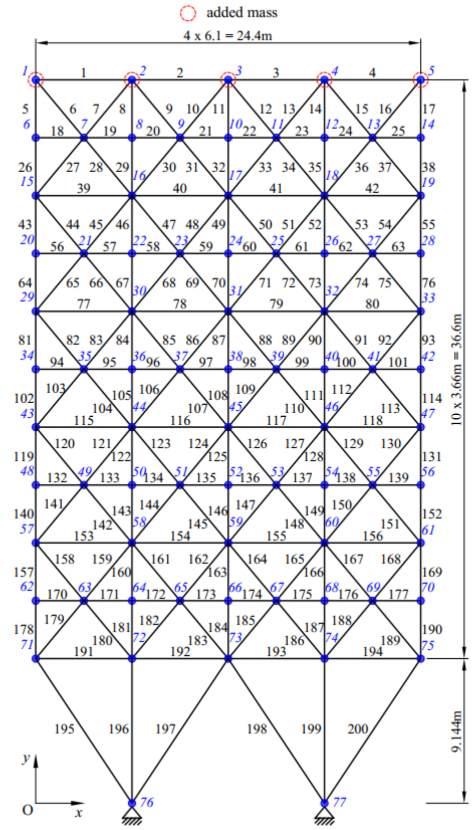

Figure 5. The 200-bar planar truss

4.4. 52-bar dome truss

A 52-bar dome truss structure as depicted in Fig. 6 is studied for shape and size optimization. For optimization design, all elements of the structure are separated into eight variable groups. A concentrated mass of 50 kg is added to each free node. Three coordinates (x, y, z) of each free node shift within the range $[-2, 2]$ m, and they are also treated as design variables. The symmetry of the entire structure must be maintained throughout the design process. There are 13 independent design variables in total, including 5 shape variables and 8 sizing ones. In terms of sizing variables, they must satisfy the condition $1 \times 10^{-4} \leq A \leq 1 \times 10^{-3}$.

A comparison between the optimal results obtained by ADE and other reference solutions is then given in Table 8. In this case, the best weight gained by ADE is close to those of the DE, and it is better than the others. The efficiency of the present method is demonstrated through the number of FE analyses whereas ADE requires only 12660 analyses (33760 analyses for DE, 20000 analyses for HS,

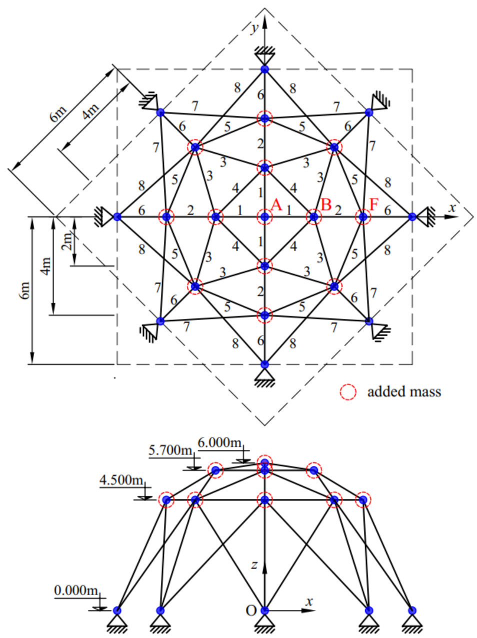

Figure 6. The 52-bar dome truss

Table 6. Optimized designs for 200-bar truss gained by the algorithms

Design variables Group A_i (cm^2)	DE	CSS-BBBC [32]	ANDE [8]	AOA	ADE
1	0.3005	0.2934	0.2912	0.3290	0.3048
2	0.4516	0.5561	0.4687	0.3910	0.4598
3	0.1000	0.2952	0.1003	0.1000	0.1000
4	0.1000	0.1970	0.1001	0.1000	0.1000
5	0.5139	0.8340	0.5531	0.5709	0.5075
6	0.8193	0.6455	0.8104	0.8151	0.8207
7	0.1001	0.1770	0.1000	0.1000	0.1001
8	1.4273	1.4796	1.4112	1.5098	1.4204
9	0.1000	0.4497	0.1000	0.1000	0.1000
10	1.5978	1.4556	1.5616	1.6406	1.5620
11	1.1590	1.2238	1.1891	1.1534	1.1583
12	0.1298	0.2739	0.1273	0.1050	0.1274
13	2.9740	1.9174	2.9704	3.0497	2.9828
14	0.1000	0.1170	0.1177	0.3752	0.1000
15	3.2598	3.5535	3.2731	4.2335	3.2612
16	1.5850	1.3360	1.5934	1.4906	1.5791
17	0.2566	0.6289	0.2037	0.5245	0.2555
18	5.0911	4.8335	5.0704	5.3419	5.1095
19	0.1004	0.6062	0.1489	0.1480	0.1004
20	5.4520	5.4393	5.4347	5.9477	5.4613
21	2.0973	1.8435	2.0755	2.0822	2.1078
22	0.6999	0.8955	0.4983	0.6488	0.6722
23	7.6705	8.1759	7.4376	5.9817	7.6301
24	0.1001	0.3209	0.1342	0.1554	0.1019
25	7.9648	10.9800	7.7775	6.9732	7.9284
26	2.8039	2.9489	2.6526	2.6465	2.7951
27	10.5086	10.5243	10.6951	15.4460	10.5555
28	21.3040	20.4271	21.9196	20.3103	21.3836
29	10.6992	19.0983	10.2209	11.6880	10.5765
Best weight (kg)	2160.6879	2298.6100	2158.8010	2217.9580	2160.7263
No. FE analysis	39500	-	11004	40000	19680
Worst weight (kg)	2160.8947	-	2302.6365	18646.8667	2161.0366
Average weight (kg)	2160.7168	-	2178.1891	7362.1928	2160.8514
Standard deviation	0.0371	-	33.4775	5976.6471	0.0946

Table 7. The first six optimal frequencies of the 200-bar truss gained by the algorithms

Frequency number	DE	CSS-BBBC [32]	ANDE [8]	AOA	ADE
1	5.000	5.010	5.000	5.000	5.000
2	12.196	12.911	12.176	12.453	12.231
3	15.026	15.416	15.116	15.022	15.038
4	16.695	17.033	16.645	17.107	16.683
5	21.369	21.426	21.308	21.283	21.422
6	21.419	21.613	-	21.798	21.437

Table 8. Optimized designs for 52-bar truss gained by the algorithms

Design variable Z_j, X_j (m); A_i (cm^2)	DE	PSO [34]	HS [35]	IDE [9]	ANDE [8]	AOA	ADE
ZA	6.0131	5.5344	4.7374	6.0052	5.9207	6.0000	6.0202
XB	2.3019	2.0885	1.5643	2.3004	2.2157	2.0000	2.2863
ZB	3.7375	3.9283	3.7413	3.7332	3.7166	4.3249	3.7457
XF	4.0000	4.0255	3.4882	4.0000	3.9344	4.0000	3.9999
ZF	2.5000	2.4575	2.6274	2.5000	2.5003	2.6030	2.5000
A1	1.0000	0.3696	1.0085	1.0001	1.0000	1.0000	1.0000
A2	1.0831	4.1912	1.4999	1.0875	1.1634	1.6093	1.0952
A3	1.2013	1.5123	1.3948	1.2135	1.2387	1.3902	1.2148
A4	1.4416	1.5620	1.3462	1.4460	1.4460	1.1975	1.4242
A5	1.4203	1.9154	1.6776	1.4315	1.3914	1.2543	1.4251
A6	1.0000	1.1315	1.3704	1.0000	1.0008	1.0000	1.0000
A7	1.5661	1.8233	1.4137	1.5623	1.6132	1.8336	1.5693
A8	1.3840	1.0904	1.9378	1.3724	1.3566	1.5711	1.3738
Best weight (kg)	193.1898	228.3810	214.9400	193.2085	193.2418	211.8887	193.1999
No. FE analysis	33760	-	20000	11040	6260	40000	12660
Worst weight (kg)	202.2523	-	-	202.4215	214.0881	4121.7596	202.5693
Average weight (kg)	197.1309	234.3000	229.8800	196.0478	200.1415	1657.1428	197.5476
Standard deviation	4.5424	5.2200	12.4400	4.1823	4.85764	1474.6922	4.5636

Table 9. The first five optimal frequencies of the 52-bar truss gained by the algorithms

Frequency number	DE	PSO [34]	HS [35]	IDE [9]	ANDE [8]	AOA	ADE
1	11.635	12.751	12.2222	11.603	11.292	14.965	11.543
2	28.648	28.649	28.6577	28.648	28.649	28.649	28.648
3	28.648	28.649	28.6577	28.648	28.649	28.649	28.648
4	28.648	28.803	28.6618	28.649	28.651	28.649	28.648
5	28.649	29.230	30.0997	28.653	28.663	29.293	28.649

and 40000 analyses for AOA). In comparison to IDE and ANDE, the current method requires more evaluations, but the optimal solution gained by ADE is superior to those of IDE and ANDE. Furthermore, ADE is more stable than ANDE, with a lower standard deviation. The optimal shape achieved by ADE for 52-bar dome truss structure is depicted in Fig. 7. The first five optimal frequencies of this structure obtained by algorithms are tabulated in Table 9 with no violation.

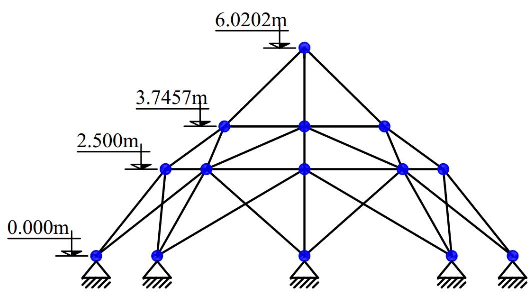

Figure 7. The optimal shape of the 52-bar truss gained using ADE

Through all the numerical examples, it can be seen that the outstanding performance of the developed ADE. ADE can maintain a balance between computational cost and quality of solution.

5. Conclusions

In this paper, we have presented a new computational approach named as ADE, which is based on a hybrid arithmetic optimization algorithm (AOA) associated with differential evolution (DE) for solving optimization problems of truss structures subjected to frequency constraints. In ADE algorithm, AOA with several modifications is performed to reinforce exploration ability whilst DE with DE/best/1 operator is used to enhance exploitation ability. Therefore, the balance between exploration and exploitation is always maintained in ADE. The improvements related to the two parameters MOA and MOP can be applied to other algorithms to make them more flexible and user-friendly. In addition, the discovery related to the limitation of exploration phase of the original AOA algorithm can be utilized in further studies on improving the AOA algorithm. From the numerical results, it is clear that ADE has been proven to be an effective tool, not only saving the computational costs but also guaranteeing the quality of gained solutions. When combined with other algorithms, this can yield even more impressive results. Moreover, the proposed algorithm can be used as an option for users for solving optimization problems. Besides, the efficiency of the algorithm on real-world constraint problems (CEC 2020) will be studied in the future.

Acknowledgements

This research is funded by Vietnam Ministry of Education and Training (MOET) under grant number B2021-VGU-04. We would like to thank Duy Tan University for their support.

References

[1] Anh, P. H. (2016). Truss sizing optimization using enhanced differential evolution with opposition-based mutation and nearest neighbor comparison. *Journal of Science and Technology in Civil Engineering (STCE)-HUCE*, 10(5):3–10.

[2] Do, D. T. T., Lee, J. (2017). A modified symbiotic organisms search (mSOS) algorithm for optimization of pin-jointed structures. *Applied Soft Computing*, 61:683–699.

[3] Nguyen-Van, S., Nguyen, K. T., Dang, K. D., Nguyen, N. T. T., Lee, S., Lieu, Q. X. (2021). An evolutionary symbiotic organisms search for multiconstraint truss optimization under free vibration and transient behavior. *Advances in Engineering Software*, 160:103045.

[4] Pierezan, J., dos Santos Coelho, L., Mariani, V. C., de Vasconcelos Segundo, E. H., Prayogo, D. (2021). Chaotic coyote algorithm applied to truss optimization problems. *Computers & Structures*, 242:106353.

[5] Kaveh, A., Zolghadr, A. (2018). Meta-heuristic methods for optimization of truss structures with vibration frequency constraints. *Acta Mechanica*, 229(10):3971–3992.

[6] Lieu, Q. X., Do, D. T. T., Lee, J. (2018). An adaptive hybrid evolutionary firefly algorithm for shape and size optimization of truss structures with frequency constraints. *Computers & Structures*, 195:99–112.

[7] Degertekin, S., Bayar, G. Y., Lamberti, L. (2021). Parameter free Jaya algorithm for truss sizing-layout optimization under natural frequency constraints. *Computers & Structures*, 245:106461.

[8] Pham, H. A. (2016). Truss optimization with frequency constraints using enhanced differential evolution based on adaptive directional mutation and nearest neighbor comparison. *Advances in Engineering Software*, 102:142–154.

[9] Ho-Huu, V., Vo-Duy, T., Luu-Van, T., Le-Anh, L., Nguyen-Thoi, T. (2016). Optimal design of truss structures with frequency constraints using improved differential evolution algorithm based on an adaptive mutation scheme. *Automation in Construction*, 68:81–94.

[10] Khot, N. S., Berke, L., Venkayya, V. B. (1979). Comparison of Optimality Criteria Algorithms for Minimum Weight Design of Structures. *AIAA Journal*, 17(2):182–190.

[11] Kaveh, A., Kalatjari, V. (2003). Topology optimization of trusses using genetic algorithm, force method and graph theory. *International Journal for Numerical Methods in Engineering*, 58(5):771–791.

[12] Sedaghati, R. (2005). Benchmark case studies in structural design optimization using the force method. *International Journal of Solids and Structures*, 42(21-22):5848–5871.

[13] Rajeev, S., Krishnamoorthy, C. S. (1992). Discrete Optimization of Structures Using Genetic Algorithms. *Journal of Structural Engineering*, 118(5):1233–1250.

[14] Storn, R., Price, K. (1997). Differential Evolution - A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11(4):341–359.

[15] Yang, X.-S. (2012). Flower Pollination Algorithm for Global Optimization. In *Unconventional Computation and Natural Computation*, Springer Berlin Heidelberg, 240–249.

[16] Phat, L. T., Long, N. N. P., Son, N. H., Vinh, H. H., Thang, L. A. (2020). Global optimization of laminated composite beams using an improved differential evolution algorithm. *Journal of Science and Technology in Civil Engineering (STCE) - HUCE*, 14(1):54–64.

[17] Son, P. V. H. (2021). Optimization of construction site layout using dynamic hybrid bacterial and ant colony algorithm. *Journal of Science and Technology in Civil Engineering (STCE) - HUCE*, 15(3):44–54.

[18] Truong, V.-H., Hung, H. M., Anh, P. H., Hoc, T. D. (2020). Optimization of steel moment frames with panel-zone design using an adaptive differential evolution. *Journal of Science and Technology in Civil Engineering (STCE) - HUCE*, 14(2):65–75.

[19] Abualigah, L., Diabat, A., Mirjalili, S., Elaziz, M. A., Gandomi, A. H. (2021). The Arithmetic Optimization Algorithm. *Computer Methods in Applied Mechanics and Engineering*, 376:113609.

[20] Agushaka, J. O., Ezugwu, A. E. (2021). Advanced arithmetic optimization algorithm for solving mechanical engineering design problems. *PLOS ONE*, 16(8):e0255703.

[21] Zhang, M., Yang, J., Ma, R., Du, Q., Rodriguez, D. (2021). Prediction of small-scale piles by considering lateral deflection based on Elman Neural Network — Improved Arithmetic Optimizer algorithm. *ISA Transactions*.

[22] Xu, Y.-P., Tan, J.-W., Zhu, D.-J., Ouyang, P., Taheri, B. (2021). Model identification of the Proton Exchange Membrane Fuel Cells by Extreme Learning Machine and a developed version of Arithmetic Optimization Algorithm. *Energy Reports*, 7:2332–2342.

[23] Njock, P. G. A., Shen, S.-L., Zhou, A., Modoni, G. (2021). Artificial neural network optimized by differential evolution for predicting diameters of jet grouted columns. *Journal of Rock Mechanics and Geotechnical Engineering*, 13(6):1500–1512.

[24] Alawad, S. M., Khalifa, A. E., Abido, M. A., Antar, M. A. (2021). Differential evolution optimization of water gap membrane distillation process for water desalination. *Separation and Purification Technology*, 270:118765.

[25] Rech, G. L., Martinotto, A. L., Balzaretti, N. M., Perottoni, C. A. (2021). Fitting of interatomic potentials by a differential evolution algorithm. *Computational Materials Science*, 187:109929.

[26] Hieu, N. T., Tuan, V. A. (2018). Weight optimization of composite cellular beam based on the differential evolution algorithm. *Journal of Science and Technology in Civil Engineering (STCE) - HUCE*, 12(5): 28–38.

[27] Anh, P. H., Duong, T. T. (2019). Weight optimisation of functionally graded beams using modified differential evolution. *Journal of Science and Technology in Civil Engineering (STCE) - HUCE*, 13(2): 48–63.

[28] Zhang, S. X., Chan, W. S., Tang, K. S., Zheng, S. Y. (2021). Adaptive strategy in differential evolution via explicit exploitation and exploration controls. *Applied Soft Computing*, 107:107494.

[29] Huynh, T. N., Do, D. T. T., Lee, J. (2021). Q-Learning-based parameter control in differential evolution for structural optimization. *Applied Soft Computing*, 107:107464.

[30] Tan, Z., Li, K. (2021). Differential evolution with mixed mutation strategy based on deep reinforcement learning. *Applied Soft Computing*, 111:107678.

[31] Wolpert, D., Macready, W. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82.

[32] Kaveh, A., Zolghadr, A. (2012). Truss optimization with natural frequency constraints using a hybridized CSS-BBBC algorithm with trap recognition capability. *Computers & Structures*, 102-103:14–27.

[33] Das, S., Abraham, A., Chakraborty, U. K., Konar, A. (2009). Differential Evolution Using a Neighborhood-Based Mutation Operator. *IEEE Transactions on Evolutionary Computation*, 13(3):526–553.

[34] Gomes, H. M. (2011). Truss optimization with dynamic constraints using a particle swarm algorithm. *Expert Systems with Applications*, 38(1):957–968.

[35] Miguel, L. F. F., Miguel, L. F. F. (2012). Shape and size optimization of truss structures considering dynamic constraints through modern metaheuristic algorithms. *Expert Systems with Applications*, 39 (10):9458–9467.