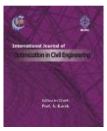
INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING
Int. J. Optim. Civil Eng., 2024; 14(3):355-383



IMPROVED ARTIFICIAL RABBITS OPTIMIZATION ALGORITHM FOR DESIGN OPTIMIZATION OF TRUSS STRUCTURES BY CONSIDERING DISCRETE DESIGN VARIABLES

S. L. SeyedOskouei, R. Sojoudizadeh[*, †], R. Milanchian, and H. Azizian
Department of Civil Engineering, Mahabad Branch, Islamic Azad University, Mahabad, Iran

[DOI: 10.22068/ijoce.2024.14.3.593]

ABSTRACT

The optimal design of structural systems represents a pivotal challenge, striking a balance between economic efficiency and safety. There has been a great challenge in balancing between the economic issues and safety factors of the structures over the past few decades; however, development of high-speed computing systems enables the experts to deal with higher computational efforts in designing structural systems. Recent advancements in computational methods have significantly improved our ability to address this challenge through sophisticated design schemes. The main purpose of this paper is to develop an intelligent design scheme for truss structures in which an optimization process is implemented into this scheme to help the process reach lower weights for the structures. For this purpose, the Artificial Rabbits Optimization (ARO) algorithm is utilized as one of the recently developed metaheuristic algorithms which mimics the foraging behaviour of the rabbits in nature. In order to reach better solutions, the improved version of this algorithm is proposed as I-ARO in which the well-known random initialization process is substituted by the Diagonal Linear Uniform (DLU) initialization procedure. For numerical investigations, 5 truss structures 10, 25, 52, 72, and 160 elements are considered in which stress and displacement constraints are determined by considering discrete design variables. By conducting 50 optimization runs for each truss structure, it can be concluded that the I-ARO algorithm is capable of reaching better solutions than the standard ARO algorithm which demonstrates the effects of DLU in enhancing this algorithm's search behaviour.

Keywords: Truss optimization; Artificial Rabbits Optimization (ARO); Metaheuristic Algorithm; Structural optimization; Diagonal Linear Uniform Initialization.

[*]Corresponding author: Department of Civil Engineering, Mahabad Branch, Islamic Azad University, Mahabad, Iran
[†]E-mail address: Reza.SojoudiZadeh@iau.ac.ir (R. Sojoudizadeh)

Received: 10 April 2024; Accepted: 4 June 2024

1. INTRODUCTION

The design process of structural systems deals with the purpose of providing an economic design scheme with a proper levels of load bearing capacity which is resistant against applied forces on the structure. The design of a structure is to achieve the goals of safety, desired performance and reliability while all design methods have been compiled by valid regulations. As one of the most important aspects of structural design, safety refers to the resistance of the structure in dealing with the effects of loads while desired performance concerns the comfort of the residence so that there should not be excessive cracking and deformation in the structural members. Finally, the reliability of the structure refers to the fact that the materials of the structure should maintain their quality throughout the expected lifetime, so that the safety and operability of the structure does not decrease too much due to aging, corrosion and other damaging factors. Among these factors, other issues like economic concern can also be of great importance while the desired performance of the an structure can be guaranteed by considering many design patterns that can be used for final construction so that finding the more economic design that satisfies the requirements of the project is one of the challenging factors in recent years. In other words, optimum design process is the way of reaching from a technological design procedure to an intelligent engineering design process by conducting multiple analysis and design procedures I order to reach a proper design scheme.

Based on the Gomez, et al. definition [1], the optimization is the process of “doing the most with the least” while the other experts like Lockhart and Johnson [2] described this process as an operation for reaching “the most favorable or effective value or condition”. In general, reaching the “best” design by taking into account some predefined specifications and criteria is the main aim of optimization. Many efforts have been conducted in optimum design of engineering problems with different sorts of optimization algorithms including the metaheuristics [3, 4]. Metaheuristic optimization algorithms are some types of intelligent approaches that can be used for leading search procedures in order to reach a most possible optimal solution which is partially near the exact global optimum point. Based on the fact that optimum design of structural engineering has received great attention due to the development of search techies like metaheuristics [5, 6], the applicability of these methods in dealing with truss optimization process is considered in this paper. By referring to the recent literature in this field, Stolpe [7] conducted a literature review on different methods of optimization applied for optimum design of truss structures by considering discrete design variables. Jiang et al. [8] proposed an improved version of the well-known whale algorithm for optimum design of truss structures. Tang and Lee [9] utilized the chaos theory to enhance he search capability of the teaching-based evolutionary algorithms for optimum design of truss structures discrete variables. Astudillo et al. [10] conducted a research on size optimization of truss bridges by measn of an enhanced firefly algorithm. Altay et al. [11] proposed the upgraded version of the salp swarm algorithm for optimum design of planar truss structures. Goodarzimehr et al. [12] utilized the Bonobo optimizer for optimization of truss structures with static constraints. Mejías et al. [13] conducted a simultaneous topology

[DOI: 10.22068/ijoce.2024.14.3.593]

optimization process by means of continuous and discrete design variables for multiresolution problems. Mashru et al. [14] investigated multi objective optimization of truss structures by proposing the multi objective version of thermal exchange algorithm. Shahrouzi and Salehi [15] proposed a new adaptive strategy for discrete optimization of space truss problems by focusing on accelerating the search process. Sellami [16] utilized the multi-stage descent algorithm for optimization of large scale truss structures in which both continuous and discrete design variables were used. Sheng-Xue [17] investigated the applicability of the medalist learning algorithm in optimum design of truss problems with frequency constraints. Khodadadi et al. [18] developed a comparative investigation to control the capability of 8 different metaheuristic algorithms for design optimization of truss problems with different sorts of static constraints. Kale et al. [19] used cohort intelligence for developing a constrained truss optimization procedure to reach economic truss designs. Chen et al. [20] investigated the structural design optimization by using a new model based on truss-continuum methodology in order to reduce the overall weight of the structures.

Recent advancements in optimization algorithms have significantly impacted the field of engineering design, offering novel and efficient solutions for various complex problems. Ranjbarzadeh et al. [21] and Seyedzadeh et al. [22] both emphasize the effectiveness of multi-objective metaheuristics in optimizing truss structures and wind farm layouts, underscoring the versatility of these approaches across different engineering challenges. In the automotive and electric vehicle sectors, Saba et al. [23] and Amini et al. [24] demonstrate the use of lattice structures and hybrid algorithms for the optimum design of components, highlighting the industry's shift towards more sustainable and efficient manufacturing methods. The introduction of chaos theory into optimization algorithms by Li and Zhang [25] and Heidari et al. [26] has opened new avenues for solving real-world engineering problems, showcasing the potential for innovative algorithmic strategies to enhance problem-solving capabilities. Moreover, the development and application of nature-inspired algorithms, as seen in the work of Shariat Panahi et al. [27], Gharibzadeh et al. [28], and Asadian et al. [29] for optimizing heat exchangers and engineering designs, reflect a growing trend towards leveraging biological processes and behaviors to inform algorithmic logic. Wang and Zhang's [30] exploration of the cheetah optimization algorithm further illustrates the ongoing exploration and application of animal-inspired algorithms in engineering optimization, particularly in the design of heat exchangers. These studies collectively indicate a robust and diverse trajectory of research within engineering optimization, showcasing the integration of multi-disciplinary approaches to address complex engineering challenges effectively.

The main purpose of this paper is to utilize one of the recently developed metaheuristics algorithm for optimum design of truss structures while a new variant of this method is also proposed in order to enhance the overall capability of the standard approach. For this purpose, the Artificial Rabbits Optimization (ARO) algorithm [31] is utilized as one of the recently developed metaheuristic algorithms which mimics the foraging behaviour of the rabbits in nature. In order to reach better solutions, the improved version of this algorithm is proposed as I-ARO in which the well-known random initialization process is substituted by the Diagonal Linear Uniform (DLU) [32] initialization procedure. For numerical investigations, 5 truss structures 10, 25, 52, 72, and 160 elements are considered in which stress and displacement constraints are determined by considering discrete design variables.

[DOI: 10.22068/ijoce.2024.14.3.593]

By conducting 50 optimization runs for each truss structure, the comparative statistical results are calculated and the performance of the ARO and I-ARO are compared to the most competitive results of other metaheuristics from the literature.

This paper introduces a significant advancement in the field of structural engineering optimization through the development and application of the I-ARO algorithm, incorporating a novel approach by substituting the traditional random initialization methods with the DLU initialization procedure for the first time. This innovative integration is specifically applied to the optimization of truss structures, encompassing a diverse range from 10 to 160 elements, with the aim of achieving the lowest possible weight while adhering to stress and displacement constraints. The research highlights the I-ARO algorithm's superior performance over existing methodologies in optimizing truss structures, demonstrated through impressive statistical outcomes across various configurations. The paper's novelty lies in this unique application of the DLU initialization in conjunction with the I-ARO algorithm, showcasing its potential to redefine optimization practices in structural engineering and beyond, hinting at future applications in diverse engineering challenges.

[DOI: 10.22068/ijoce.2024.14.3.593]

2. PROBLEM STATEMENT OF DISCRETE TRUSS OPTIMIZATION

The main focus of this section is on formulating a structural design optimization problem aimed at minimizing the weight of truss structures by considering specific design constraints. The main objective is to reduce the overall weight of the truss structures by using discrete design variables that assign predefined design sections to the structural elements during the optimization process. The mathematical representation of these aspects is as follows while the predefined set of discrete cross-sectional areas is denoted by $\boldsymbol{S}$:

$$Weight\ (\boldsymbol{A}) = \sum_{i=1}^{e} \rho_i\, I_i\, A_i\,. \qquad i = 1.2.\ldots.e, \tag{1}$$

$$\boldsymbol{A} \in \boldsymbol{S} = \{A_1\,.A_2\,.\ldots.A_i\} \tag{2}$$

where the vector $\boldsymbol{A}$ contains the cross-sectional area of the design sections (A_i); ρ_i represents the density of the steel material; I_i denotes the length of the structural elements; e is the total number of structural elements in the structure.

The design constraints are formulated by considering the nodal displacement of the truss structures and the stress in structural elements as follows:

$$\delta_{min} \le \delta_j \le \delta_{max}. \qquad j = 1.2.\ldots.n \tag{3}$$

$$\sigma_{min} \le \sigma_i \le \sigma_{max}. \qquad i = 1.2.\ldots.e \tag{4}$$

where δ_j and σ_i represent the nodal displacement in *jth* node and elemental stress in *ith* element; n is the total number of structural nodes; δ_{min} and σ_{min} are the lower bound of the constraints; δ_{max} and σ_{max} are the upper bound of the constraints; n is the total number of

structural nodes in the structure.

Since the truss optimization problem has a set of constraints to be taken care of, a proper constraint handling approach is formulated as follows which is a type of penalty handling approach:

$$f_{penalty}(\boldsymbol{A}) = (1 + \varepsilon_1, v)^{\varepsilon_2} \times Weigth\,(\boldsymbol{A}) \tag{5}$$

$$v = \sum_{i=1}^{h} max\{0. g_i(\boldsymbol{A})\} \tag{6}$$

where v represents the summation of the violated design constraints; h is the total number of design constraints; $g_i(\boldsymbol{A})$ represents the *ith* design constraint; ε_1 and ε_2 are used to control the penalty applied to the constraints.

3. ARTIFICIAL RABBITS OPTIMIZATION (ARO) ALGORITHM

In this section, the overall description of the ARO algorithm is presented by focusing on inspirational concept and the mathematical model of the algorithm.

The ARO was inspired by the survival strategies of rabbits in their natural habitat. These strategies have evolved over time as a means of ensuring the rabbits' survival and avoiding predators. One such strategy involves the rabbits' behavior of not eating the grass near their nests to prevent detection by predators. Instead, they venture further away to find food, utilizing their wide field of vision for overhead scanning. Another survival strategy employed by rabbits is random hiding. To escape predators or hunters, rabbits create multiple burrows around their nest and randomly choose one as a shelter (Fig. 1). Their physical attributes, such as short forelegs and long back legs, along with strong muscles and tendons, enable them to run at high speeds [31]. This strategy is utilized in the main search loop of the ARO algorithm as exploitation phase while the former strategy is considered in the exploration phase.

Figure 1. A Rabbit with multiple nests in nature [31]

In the first stage of the ARO algorithm, a random initialization process is conducted as

[DOI: 10.22068/ijoce.2024.14.3.593]

follows in order to determine the initial position of the search agents:

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} x_1^1 & x_1^2 & \cdots & x_1^j & \cdots & x_1^d \\ x_2^1 & x_2^2 & \cdots & x_2^j & \cdots & x_2^d \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ x_i^1 & x_i^2 & \cdots & x_i^j & \cdots & x_i^d \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ x_n^1 & x_n^2 & \cdots & x_n^j & \cdots & x_n^d \end{bmatrix}, \quad \begin{cases} i = 1,2, \dots, n. \\ j = 1,2, \dots, d. \end{cases} \tag{7}$$

$$x_i^j = x_{i,min}^j + rand.\left(x_{i,max}^j - x_{i,min}^j\right), \quad \begin{cases} i = 1,2, \dots, n. \\ j = 1,2, \dots, d. \end{cases} \tag{8}$$

where x_i is position vector of the *ith* rabbit; n and d refer to the rabbits' total population and dimension of the optimization problem respectively; $x_{i,max}^j \; and \; x_{i,min}^j$ relates to the upper and lower bounds of the optimization variables; $rand$ denotes to a random number in the range of 0 and 1.

The main loop of the ARO algorithm is developed based on the previously mentioned two surviving strategies of rabbits in nature including the detour foraging which is utilized for exploration phase of the algorithm and random hiding behaviour for exploitation phase. For this purpose, each rabbit in the swarm is assigned its own region containing some grass and several burrows. During foraging, rabbits visit the positions of other rabbits in the swarm randomly. They tend to move around a food source, adding a perturbation to their movement to ensure they gather enough food. The mathematical representation of this detour foraging behaviour in ARO involves each search individual updating its position towards another randomly chosen search individual within the swarm, while incorporating a perturbation to their movement. The mathematical model of this phase in the ARO algorithm is developed as follows:

$$v_i(t+1) = x_j(t) + R.\left(x_i(t) - x_j(t)\right) + round\left(0.5.\left(0.05 + r_1\right)\right).n_1, \quad i = 1,2, \dots, n \tag{9}$$

$$R = L.c \tag{10}$$

$$L = \left(e - e^{\left(\frac{t-1}{T}\right)^2}\right).\sin\left(2\pi r_2\right) \tag{11}$$

$$c(k) = \begin{cases} 1 & if \; k == g(l) \\ 0 & else \end{cases} \qquad \begin{matrix} k = 1, \dots, d \\ l = 1, \dots, \lceil r_3 \cdot d \rceil \end{matrix} \tag{12}$$

$$g = randperm(d) \tag{13}$$

[DOI: 10.22068/ijoce.2024.14.3.593]

$$n_1 \sim N(0, 1) \tag{14}$$

where $v(t+1)$ denotes new position of the *ith* rabbit; T is the total number of optimization iterations; $x_i(t)$ *and* $x_j(t)$ denote the *ith* and *jth* rabbits' position at current iteration; *randperm* generates integer random numbers between 1 and d; $r1$, $r2$, and $r3$ refers to random numbers of range 0 and 1.

In Fig. 2, the variation of the running length in rabbits (*L)* is depicted, which indicates the distance covered during detour foraging while in Fig. 3, the variation of R as running operator is illustrated which follows a standard normal distribution.

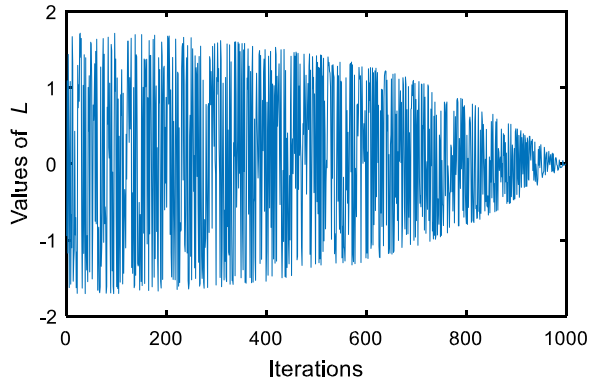


Figure 2. Changing in the *L* values during time [31]

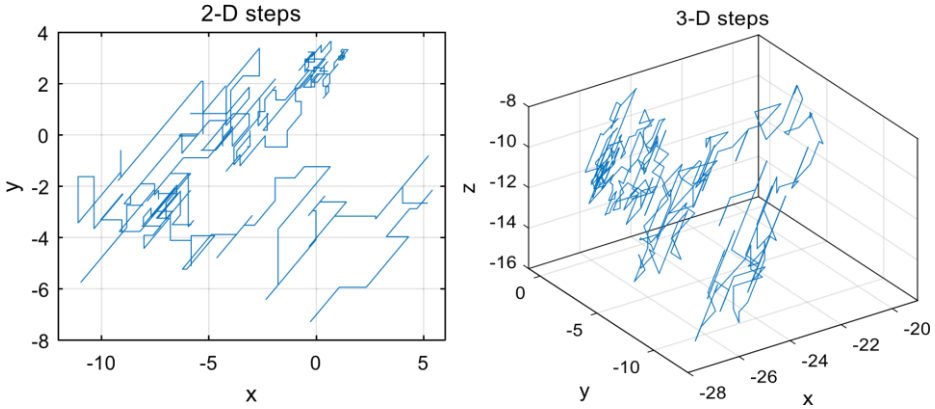


Figure 3. Changing in the *R* values during time [31]

For evading the predators in the exploitation phase, burrows creation process is conducted which is modeled by generating *d* new vectors around the current position of the rabbits. The rabbits selects one of these burrows randomly to reduce the predation risk. The equation provided below describes the generation of the *j*th burrow for the *i*th rabbit:

$$b_{i,j}(t) = x_i(t) + H.g.x_i(t), \qquad \begin{cases} i = 1,2,\dots,n. \\ j = 1,2,\dots,d. \end{cases} \tag{15}$$

[DOI: 10.22068/ijoce.2024.14.3.593]

$$H = \frac{T - t + 1}{T} . r_4 \tag{16}$$

$$\text{n2}\sim\text{N}(0,1) \tag{17}$$

$$\text{g}(k) = \begin{cases} 1 & if\ k == \text{j} \\ 0 & else \end{cases} \qquad k = 1, \dots, d \tag{18}$$

where H refers to a parameter which denotes on hiding with a linear decrease from 1st iteration to *Tth* iteration.

Based on the earlier description of rabbits' behavior in nature, they often encounter the danger of predators chasing and attacking them. To ensure their survival, they need to find a safe hiding spot so, they tend to select one of their accessible burrows randomly to seek refuge and avoid being captured. To express this random hiding behavior in mathematical terms, the following equations are used:

$$v_i(t+1) = x_i(t) + R.\left(r_4.b_i(t) - x_i(t)\right), \qquad i = 1,2,\dots,n. \tag{19}$$

$$b_{i,r}(t) = x_i(t) + H.g_r.x_i(t), \qquad \begin{cases} i = 1,2,\dots,n. \\ j = 1,2,\dots,d. \end{cases} \tag{20}$$

$$\text{g}(k) = \begin{cases} 1 & if\ k == \lceil r_5 \cdot d \rceil \\ 0 & else \end{cases} \qquad k = 1, \dots, d \tag{21}$$

where $b_{i,r}$ is the burrow which is selected randomly for hiding; r_4 and r_5 are randomly generated numbers in the range of (0,1).

The position updating process of rabbits after conducting the procedures for both exploration and exploitation phases are handled as follows:

$$x_i(t+1) = \begin{cases} x_i(t) & f(x_i(t)) \le f(v_i(t+1)) \\ v_i(t+1) & f(x_i(t)) > f(v_i(t+1)) \end{cases} \tag{22}$$

In the ARO algorithm, a switch between two phases of exploration and exploitation is modeled as energy shrink (Fig. 4) in which a smooth transit is conducted by means of the following equation:

$$\text{A(t)=4(1-t/T)ln1/r} \tag{23}$$

where *r* is a randomly generated number in the range of 0 and 1; t is the current iteration and T is the maximum number of considered iterations.

[DOI: 10.22068/ijoce.2024.14.3.593]

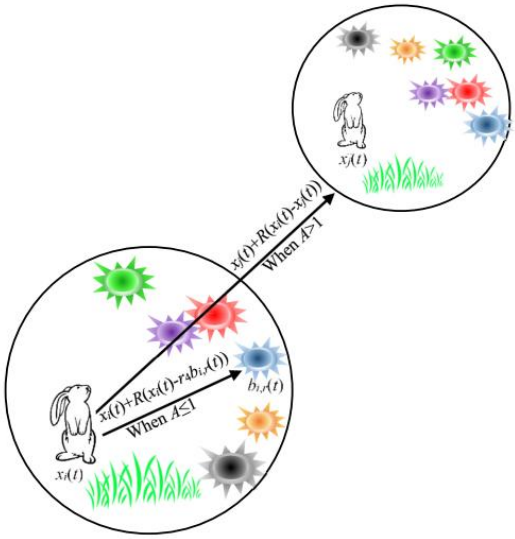


Figure 4. Energy shrink action during search process [31].

In Alg. 1 and Fig. 5, the pseudo code and the flowchart of the ARO are presented respectively.

Algorithm 1: The pseudo code of the ARO algorithm [33].

1: Initialize the parameters of ARO: n, m, and T
2: Initialization of ARO's population
    $\bar{x}i,j = \bar{x}jmin + (\bar{x}jmax - \bar{x}jmin) \times U(0,1)$ ∀i = 1,2,…,n, and ∀j = 1,2,…,m
3: Calculate f($\bar{x}i$) ∀i = 1,2,…,n {Fitness evaluation}
4: Select the best solution so far $\bar{x}best$
5: t=1
6: while (t ≤ T) do
7: for i=1 : n do
8: Calculate the energy factor A using
9: if A > 1 then
10: Select a random rabbit $\bar{x}k$, where k ≠ i
11: Calculate R using
12: Perform detour foraging using
13: else
14: Generate d burrows and randomly select one
15: Perform random hiding action using
16: end if
17: Calculate fitness of $\bar{x}i$
18: Update position of $\bar{x}i$
19: Update the $\bar{x}best$
20: end for
21: t = t + 1
22: end while
23: Return the best solution $\bar{x}best$

[DOI: 10.22068/ijoce.2024.14.3.593]

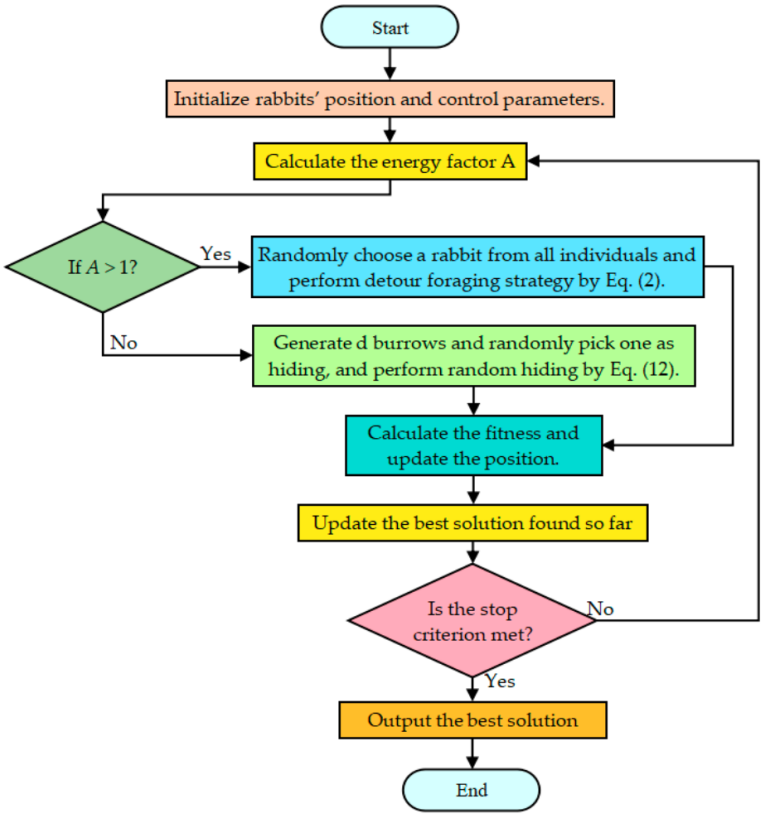


Figure 5. Flowchart of ARO algorithm [34]

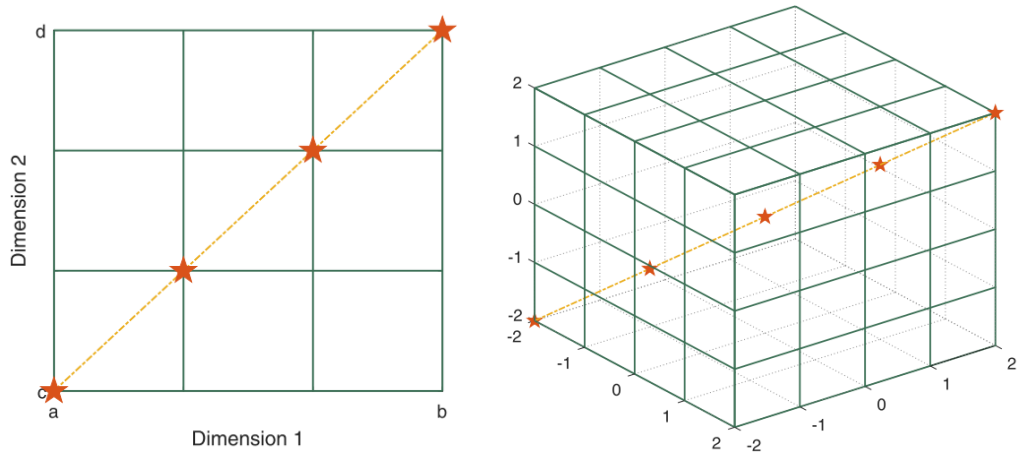
4. IMPROVED ARO (I-ARO) ALGORITHM

Random number generation has been one of the most frequent ways of creating new solution candidates in most of the metaheuristic algorithms. Random movements involve a series of consecutive random steps for various purposes. The initial position determination alongside the movements of candidates in the main loop of the algorithms can be done by using randomization process based on the concept of Brownian random motion. However, this procedure leads to poor convergence behavior by the algorithms and the possibility of entrapment in local optimal points is increased in this method. In the ARO algorithm, the Brownian random generation is utilized in different phases on the algorithm especially in the initialization part in which positon vectors are determined randomly by considering the upper and lower bounds of the variables. The initialization process has a great impact on the optimization procedure and the quality of the final global optimal points while the random initialization process concerns the diversity and uniformity of population distribution without taking into account the update mechanism of the algorithm. In this regard, there is an urgent need to develop novel techniques for enhancing the initialization process of the algorithms which can increase the searching capability of the algorithms. For this purpose, the Improved ARO (I-ARO) is proposed in this section in which the Brownian random

[DOI: 10.22068/ijoce.2024.14.3.593]

initialization process of the ARO is replaced by the a new initialization scheme called Diagonal Linear Uniform (DLU) initialization process [32].

In the DLU initialization process, the dimension of the search space are divided into equal parts of N-1 in the first phase while the vertices of the diagonal subspace are selected accordingly. In other words, some uniform points on the "diagonal" of the space are selected (Fig. 6.a) while the total distance between adjacent points is (xu-xl)/(N-1). For instance, if five initial individuals are required in a 3-dimensional space by considering the upper and lower bound vectors of (-2,-2,-2) and (2,2,2) respectively, each dimension is divided into four parts by the DLU initialization process hile the DLU method selects the five initial points as (2,2,2), (-2,-2,-2), (1,1,1), (0,0,0) and (-1,-1,-1) (Fig. 6.b). The initialization by means of DLU is a straightforward and readily applicable method. What's crucial is that its effectiveness remains consistent even when dealing with higher dimensions, and it shows strong performance across diverse problem categories, encompassing multi-objective and multimodal problems. The pseudocode for DLU initialization process is presented in Fig. 7.



(a) (b)

Figure 6. The diagrams of DLU initialization process for 2D (a) and 3D (b) spaces [32]

```
Input: the range [xi_min, xi_max] for each dimension xi, the population size N, the dimension of problem D.
X_init = zeros(N, D) // allocate memory space
for j = 1, 2, ..., N do
   for i = 2, ..., D - 1 do
      dx = (xi_max - xi_min) / (N - 1) // length of each interval in i-th dimension
      X_init(j, 1) = xi_min
      X_init(j, i) = xi_min + (i - 1) * dx
      X_init(j, D) = xi_max
   end for
end for
return X_init as a vector.
```

Figure 7. Pseudo code of the DLU approach [32]

[DOI: 10.22068/ijoce.2024.14.3.593]

5. TRUSS DESIGN EXAMPLES WITH DISCRETE VARIABLES

In this section, the details of the considered truss design examples with discrete design variables are described while the basic characteristics of the utilized material alongside the truss structures' specific details are explained. The first design example is a truss structure with 10 structural members and 6 nodes (Fig. 8). The density of the steel material is set to 0.1 lb/in^3 and the modulus of elasticity is 10^4 ksi. The displacement and stress constraints are as ±2 in. and ±25 ksi respectively while the discrete design variables are presented in Table 1.

The second design example is a truss structure with 10 nodes and 25 structural members (Fig. 9). The modulus of elasticity is 10^4 ksi and the density of the steel material is set to 0.1 lb/in^3. The displacement and stress constraints are as ±0.35 in. and ±40 ksi respectively while the discrete design variables are presented in Table 1.

The 52-bar truss structure is the third design example in this paper which is comprised of 20 node and 52 structural members (Fig. 10). The only constraint in this structure is stress limitations with allowable range of ±180 Mpa while the modulus of elasticity and the density of the material are set to 207 GPa and 7860 kg/m^3 respectively. The discrete design variables are presented in Table 1.

The 72-bar truss structure is the forth design example which is comprised of 20 node and 72 structural members (Fig. 11). The only constraint in this structure is stress limitations with allowable range of ±25 Mpa while the modulus of elasticity and the density of the material are set to 10^4 ksi and 0.1 lb/in^3 respectively. The discrete design variables are presented in Table 1.

The last design example is a 160-bar truss structure with 52 nodes and 160 structural elements (Fig. 12). The buckling stress limitations of $\sigma_b = 1300 - (kl/r)^2/24$ for $kl/r \leq 120$ and $\sigma_b = 10^7/(kl/r)^2$ for $kl/r > 120$ are considered in design process. The modulus of elasticity and density of material are 2.047×106 kg/cm^2 and 0.00785 kg/cm^3 respectively. The discrete design variables are presented in Table 1.

Table 1: The discrete design variables of the truss design examples.

Truss Structure	**Discrete Variables (in^2)**
10-bar Truss Structure	{1.62,1.80,1.99,2.13,2.38,2.62,2.63,2.88,2.93,3.09,3.13,3.38,3.47,3.55,3.63,3.84, 3.87,3.88,4.18,4.22,4.49,4.59,4.80,4.97,5.12,5.74,7.22,7.97,11.50,13.50,13.90, 14.20,15.50,16.00,16.90,18.80,19.90,22.00,22.90,26.50,30.00,33.50}
25-bar Truss Structure	{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.6, 2.8, 3.0, 3.2, 3.4}
52-bar Truss Structure	{71.613, 90.968, 126.451, 161.29 198.064, 252.258, 285.161, 363.225, 388.386, 494.193, 506.451, 641.289, 645.16, 792.256, 816.773, 939.998, 1008.385, 1045.159, 1161.288, 1283.868, 1374.191, 1535.481, 1690.319, 1696.771, 1858.061, 1890.319, 1993.544, 729.031, 2180.641, 2238.705, 2290.318, 2341.931, 2477.414, 2496.769, 2503.221, 2696.769, 2722.575, 2896.768, 2961.284, 3096.768, 3206.445, 3303.219, 3703.218, 4658.055, 5141.925, 5503.215, 5999.988, 6999.986, 7419.34, 8709.66, 8967.724, 9161.272, 9999.98,

[DOI: 10.22068/ijoce.2024.14.3.593]

	10322.56, 10903.204, 12129.008, 12838.684, 14193.52, 14774.164, 15806.42, 17096.74, 18064.48, 19354.8, 21612.86}
72-bar Truss Structure	{0.111, 0.141, 0.196, 0.25, 0.307, 0.391,0.442, 0.563, 0.602, 0.766, 0.785, 0.994, 1, 1.228, 1.266, 1.457, 1.563 1.62, 1.8 1.99, 2.13, 2.38, 2.62, 2.63, 2.88, 2.93 3.09, 1.13 3.38, 3.47, 3.55 3.63, 3.84, 3.87, 3.88, 4.1 4.22 4.49, 4.59, 4.8, 4.97 5.12, 5.74, 7.22, 7.97 8.53 9.3 10.85, 11.5, 13.5, 13.9, 14.2, 15.5, 16 16.9 18.8 19.9 22, 22.9, 24.5, 26.5, 28,30, 33.5}
160-bar Truss Structure	{1.84, 2.26, 2.66, 3.07, 3.47, 3.88, 4.79, 5.27, 5.75, 6.25, 6.84, 7.44, 8.06, 8.66, 9.40, 10.47, 11.38, 12.21, 13.79, 15.39, 17.03, 19.03, 21.12, 23.20, 25.12, 27.50, 29.88, 32.76, 33.90, 34.77, 39.16, 43.00, 45.65, 46.94, 51.00, 52.10, 61.82, 61.90, 68.30, 76.38, 90.60, 94.13 cm2}, and r = {0.47, 0.57, 0.67, 0.77, 0.87, 0.97, 0.97, 1.06, 1.16, 1.26, 1.15, 1.26, 1.36, 1.46, 1.35, 1.36, 1.45, 1.55, 1.75, 1.95, 1.74, 1.94, 2.16, 2.36, 2.57, 2.35, 2.56, 2.14, 2.33, 2.97, 2.54, 2.93, 2.94, 2.94, 2.92, 3.54, 3.96, 3.52, 3.51, 3.93, 3.92, 3.92 }

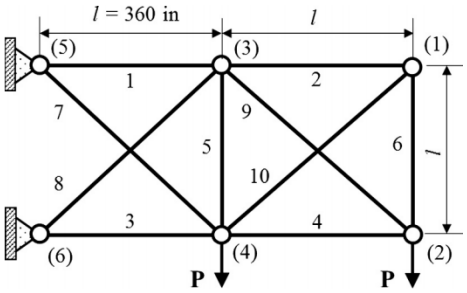


Figure 8. 10-bar truss structure

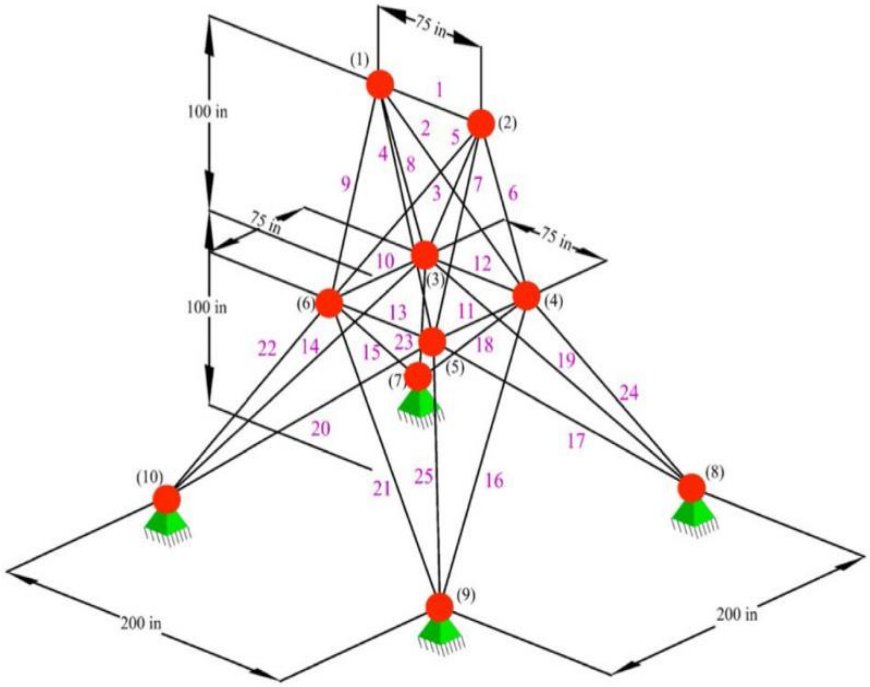


Figure 9. 25-bar truss structure

[DOI: 10.22068/ijoce.2024.14.3.593]

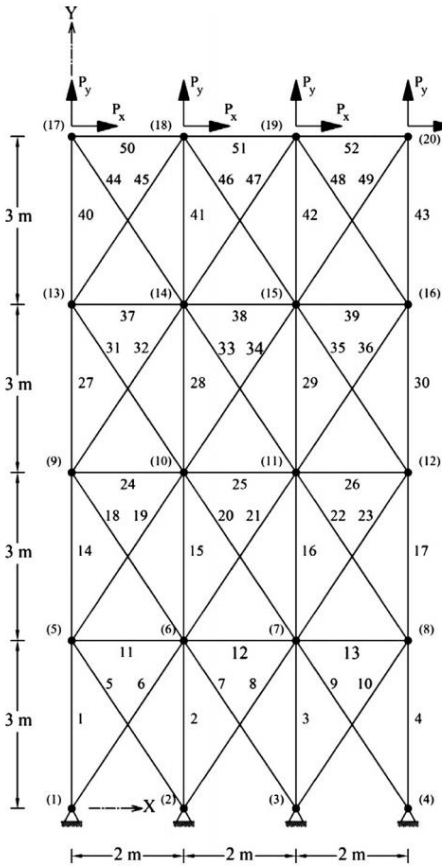


Figure 10. 52-bar truss structure

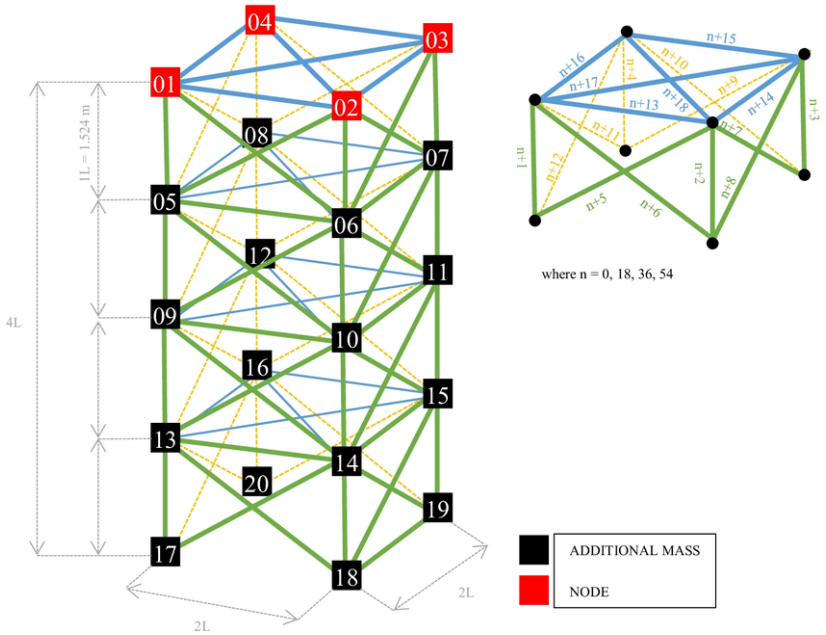


Figure 11. 72-bar truss structure

[DOI: 10.22068/ijoce.2024.14.3.593]

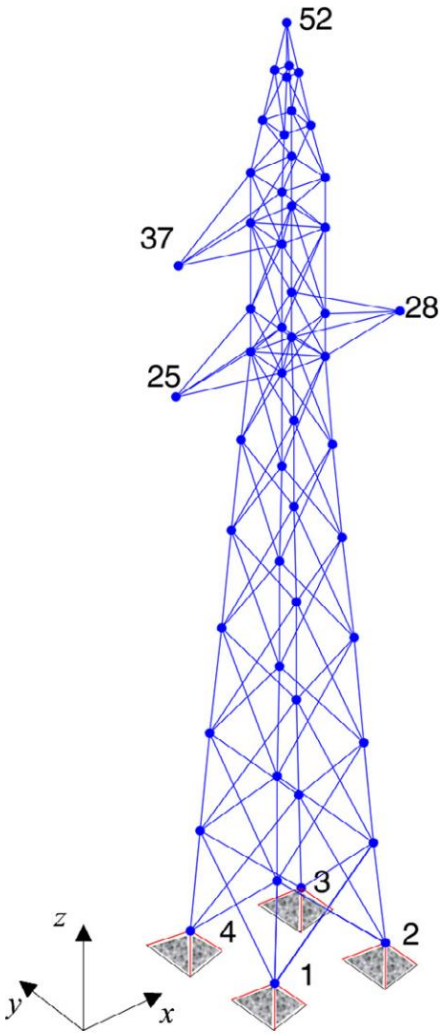


Figure 12. 160-bar truss structure

6. RESULTS OF NUMERICAL INVESTIGATIONS

In this section, the detailed results of the numerical investigations including the optimization procedures are presented.

6.1. 10-bar truss problem

Regarding the first design example which is a 10-bar truss structure, the convergence curves of the best optimization runs for the I-ARO and ARO algorithms are presented in Fig. 13 in which the superiority of the I-ARO in reaching better results than ARO is demonstrated.

[DOI: 10.22068/ijoce.2024.14.3.593]

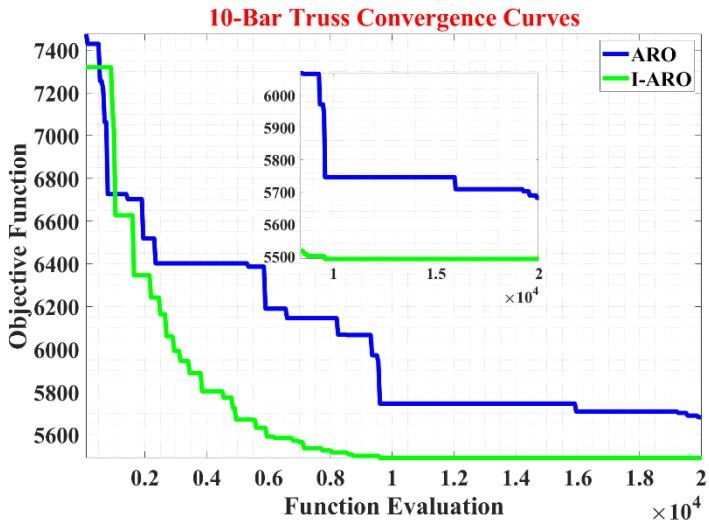


Figure 13. Convergence history of ARO and I-ARO algorithms for 10-bar truss structure.

The best result of the multiple optimization runs by the ARO and I-ARO algorithms are presented in Table 1 for the 10-bar truss problem in which the discrete design variables are also provided for comparative purposes. The lowest possible weight for the structure is calculated for the ARO and I-ARO algorithms while I-ARO can reach 5491.71 lb.

The competitive and statistical results of the I-ARO and ARO algorithms are presented in Table 2 by considering the conducted 50 independent runs alongside the results of other approaches. It can be seen that the I-ARO algorithm is capable of reaching 5491.71 lb for the weight of the 10-bar truss structure while the MBA with 5504.75 lb has the second rank. The I-ARO algorithm is capable of providing better statistical results than other methods.

Table 2: Comparative results of ARO and I-ARO algorithms and other approaches in dealing with 10-bar truss problem

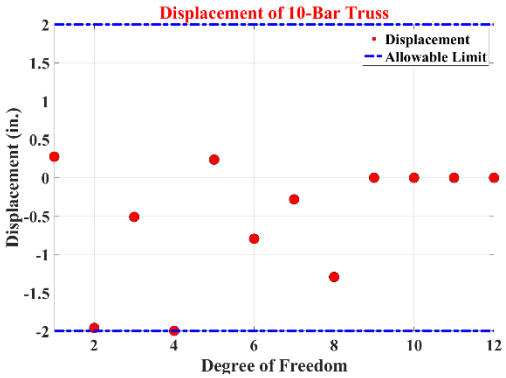
Design Variables (in.^{2})	**GA [35]**	**PSO [36]**	**PSOPC [36]**	**HPSO [36]**	**MBA [37]**	**ARO**	**I-ARO**
A_1	33.5	30	30	30	30	33.5	33.5
A_2	1.62	1.62	1.8	1.62	1.62	3.63	1.62
A_3	22	30	26.5	22.9	22.9	30	22.9
A_4	15.5	13.5	15.5	13.5	16.9	14.2	15.5
A_5	1.62	1.62	1.62	1.62	1.62	1.62	1.62
A_6	1.62	1.8	1.62	1.62	1.62	1.62	1.62
A_7	14.2	11.5	11.5	7.97	7.97	7.97	7.97
A_8	19.9	18.8	18.8	26.5	22.9	22.9	22
A_9	19.9	22	22	22	22.9	18.8	22
A_{10}	2.62	1.8	3.09	1.8	1.62	2.13	1.62
Weight (lb)	5613.84	5581.76	5593.44	5531.98	5507.75	5681.7455	5491.7174
Worst weight (lb)	–	–	–	–	5536.965	6539.7499	6250.6790
Mean	–	–	–	–	5527.296	5989.9573	5663.3019

[DOI: 10.22068/ijoce.2024.14.3.593]

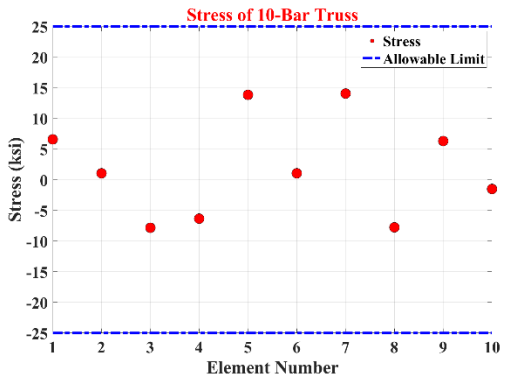
weight (lb)							
Standard deviation (lb)	–	-	-	3.8402	11.38	190.2765	169.7751

HPSO: Heuristic Particle Swarm Optimization
MBA: Mine Blast Algorithm
DE: Differential Evolution
AEDE: Adaptive Elitist Differential Evolution

The design constraints related to the best optimization run conducted by the I-ARO are presented in Fig. 14 for both displacements and stresses of the structural nodes and members. The results show that the I-ARO can handle the constraints in the allowable ranges.



(a)



(b)

Figure 14. Displacement and stress design constraints for 10-bar truss problem

6.2. 25-bar Space Structure

For the 20-bar truss structure, the convergence history of the best optimization run alongside the conducted 50 independent runs for both ARO and I-ARO are presented in Fig. 15. The I-ARO algorithm is capable of reaching better results than the standard ARO algorithm.

[DOI: 10.22068/ijoce.2024.14.3.593]

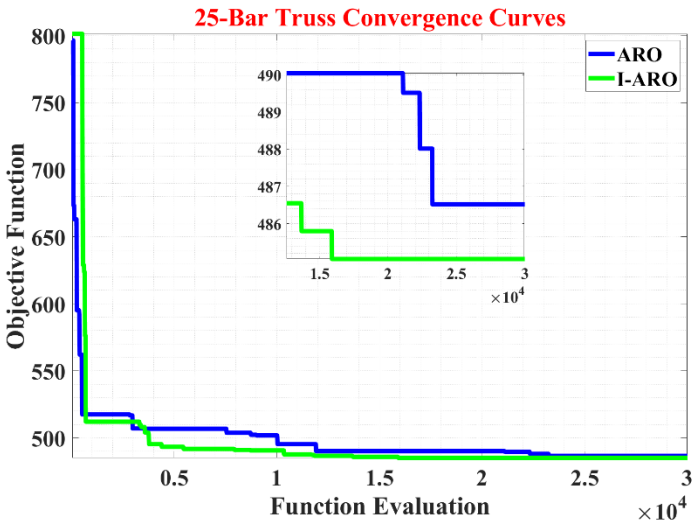


Figure 15. Convergence history of ARO and I-ARO algorithms for 25-bar truss structure.

Table 3 reports the statistical and detailed results of deferent methods for the 20-bar truss problem alongside the results of the ARO and I-ARO algorithms. The I-ARO can provide a best optimum value of 485.04 lb while the ARO with 486.51 lb has the second rank. The statistical results for the other methods are not available but the I-ARO with mean of 495.63 lb, worst of 502.63 lb and std. of 5.76 lb outranked the standard ARO algorithm for this case.

Table 3: Comparative results of ARO and I-ARO algorithms and other approaches in dealing with 25-bar truss problem

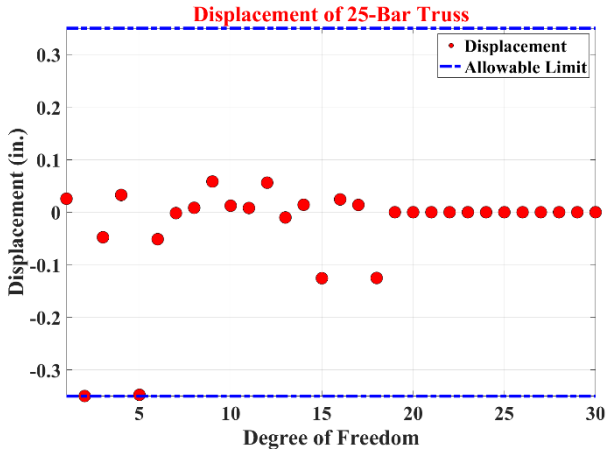
Design Variables (in.2)	**SGA [38]**	**SA [39]**	**PSO [36]**	**PSOPC [36]**	**GA [40]**	**GAOS [40]**	**ITA [41]**	**ARO**	**I-ARO**
A1	0.1	0.1	0.4	0.1	0.1	0.1	0.1	0.2	0.1
A2	0.5	1.2	0.6	1.1	1.8	1.2	1.9	0.6	0.5
A3	3.4	3.4	3.5	3.1	2.3	3.2	2.6	3.4	3.4
A4	0.1	0.1	0.1	0.1	0.2	0.1	0.1	0.1	0.1
A5	1.5	2.2	1.7	2.1	0.1	1.1	0.1	1.6	1.9
A6	0.9	1.1	1.0	1.0	0.8	0.9	0.8	1	1
A7	0.6	1.0	0.3	0.1	1.8	0.4	2.1	0.4	0.4
A8	3.4	3.0	3.4	3.5	3.0	3.4	2.6	3.4	3.4
Weight (lb)	486.29	537.23	486.54	490.16	546.01	493.80	562.93	486.519	485.0488
Worst weight (lb)	–	–	–	–	–	–	–	512.094	502.6337
Mean weight (lb)	–	–	–	–	–	–	–	496.393	495.6369

[DOI: 10.22068/ijoce.2024.14.3.593]

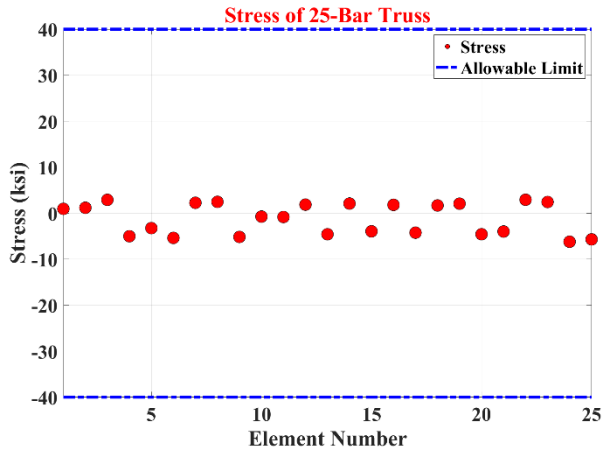
Standard deviation (lb)	–	–	–	–	–	–	–	7.6017	5.7694

SGA: Steady-State Genetic Algorithm
GA: Genetic Algorithm
PSO: Particle Swarm Optimization
PSOPC: Particle Swarm Optimization Passive Congregation
SA: Simulated Annealing
GAOS: Genetic Algorithm Based Optimum Structural Design
ITA: Improved Templeman Algorithm

The displacement and stress design constraints are reported in Fig. 16 regarding the best optimization run conducted by the I-ARO algorithm in dealing with the 25-bar truss structure for which it is capable of handling the constraints properly.



(a)



(b)

Fig. 16. Displacement and stress design constraints for 25-bar truss problem.

[DOI: 10.22068/ijoce.2024.14.3.593]

6.3. 52-bar Planar Structure

For the third design example, the convergence history of the best optimization run among 50 conducted runs are illustrated in Fig. 17 for both ARO and I-ARO algorithms while the capability of the I-ARO algorithm in providing better results than standard ARO algorithm is demonstrated.

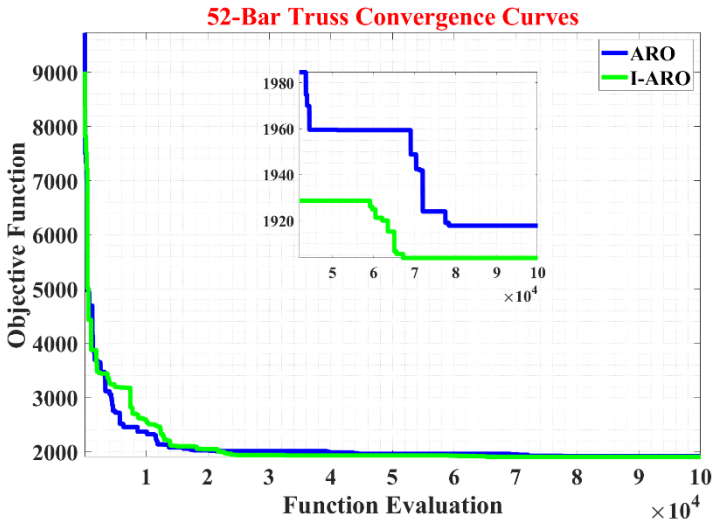


Figure 17. Convergence history of ARO and I-ARO algorithms for 52-bar truss structure.

In Table 4, the design variables related to the best optimization runs conducted by different methods including the ARO and I-ARO algorithms are presented alongside the statistical results. The I-ARO algorithm is capable of reaching 1903.70 lb for the weight of the 52-bar truss structure while the other methods converge to higher values. Regarding statistical results, the I-ARO can provide 2222.43 lb, 2859.54 lb and 269.17 lb for the mean, worst and std which are better than the results of the ARO algorithm.

Table 4: Comparative results of ARO and I-ARO algorithms and other approaches in dealing with 52-bar truss problem

Design Variables (in.2)	SGA [38]	SSGA-2P [38]	SSGA-3P [38]	PSOPC [36]	HPSO [36]	PSO [36]	HS [42]	ARO	I-ARO
A1	4658.055	4658.055	4658.055	5999.988	4658.055	4658.055	4658.055	4658.055	4658.055
A2	1161.288	1161.288	1283.868	1008.380	1161.288	1374.190	1161.288	1161.288	1161.288
A3	645.16	645.16	285.161	2696.770	363.225	1858.060	506.451	363.225	388.386
A4	3303.219	3303.219	3303.219	3206.440	3303.219	3206.440	3303.219	3303.219	3303.219
A5	1045.159	1045.159	1045.159	1161.290	940.000	1283.870	940.000	939.998	939.998
A6	494.193	494.193	363.225	729.030	494.193	252.260	494.193	641.289	729.031
A7	2477.414	2477.414	2496.769	2238.710	2238.705	3303.220	2290.318	2180.641	2238.705
A8	1045.159	1045.159	1045.159	1008.380	1008.385	1045.160	1008.385	1008.385	1008.385
A9	285.161	285.161	363.225	494.190	388.386	126.450	2290.318	494.193	388.386
A10	1696.771	1696.771	1696.771	1283.870	1283.868	2341.93	1535.481	1161.288	1283.868

[DOI: 10.22068/ijoce.2024.14.3.593]

A11 **A12**	1045.159 641.289	1045.159 641.289	1045.159 792.256	1161.290 494.190	1161.288 792.256	1008.38 1045.16	1045.159 506.451	1161.288 1161.288	1161.288 494.193
Weight (lb)	1970.142	1970.110	1980.412	2146.63	1905.49	2230.16	1906.76	1917.7879	1903.7007
Worst weight (lb)	–	–	–	–	–			3453.3942	2859.5465
Mean weight (lb)	–	–	–	–	–			2305.4600	2222.4305
Standard deviation (lb)	–	–	–	–	–			384.5965	269.1790

HS: Harmony Search Algorithm
SGA: Steady-State Genetic Algorithm
GA: Genetic Algorithm
PSO: Particle Swarm Optimization
PSOPC: Heuristic Particle Swarm Optimization Passive Congregation
HPSO: Heuristic Particle Swarm Optimization

Fig. 18 illustrates the stress design constraints for the optimal runs performed by the I-ARO algorithm. The figure showcases the effectiveness of the constraint handling approach employed by the I-ARO in this paper.

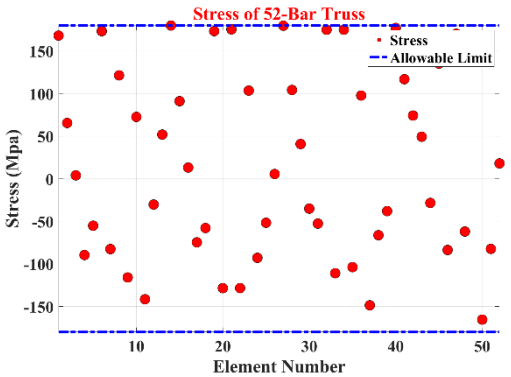


Figure 18. Stress design constraints for 52-bar truss problem

6.4. 72-bar Truss Structure

Fig. 19 displays the convergence behaviour of the ARO and I-ARO algorithms as they are utilized for addressing the 72-bar truss design problem that incorporates discrete variables. In parallel, Table 5 offers a juxtaposition of the acquired outcomes. The graphical representations in Fig. 19 distinctly indicate I-ARO's superiority over ARO, as evidenced by its accomplishment of a 389.45 lb weight, outperforming several extensively acknowledged metaheuristic algorithms. Nonetheless, statistical results underscore that I-ARO's performance remains remarkably competitive within this specific scenario.

[DOI: 10.22068/ijoce.2024.14.3.593]

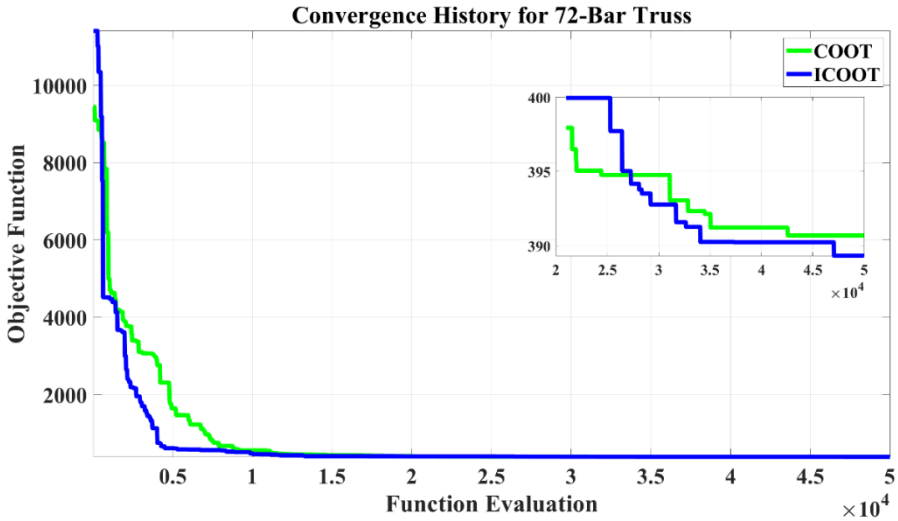


Figure 19. Convergence curves of the ARO and I-ARO algorithms regarding 72-bar truss

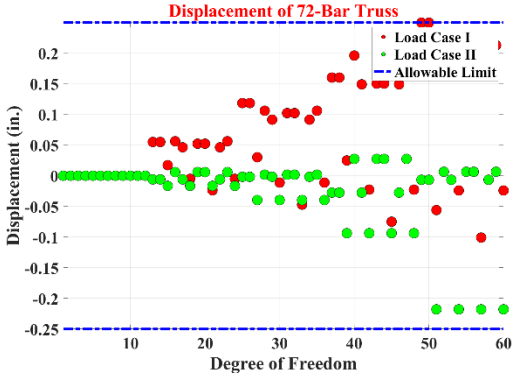
Table 5: Comparative results of different approaches in dealing with 72-bar truss problem

Design Variables (in.2)	SGA [38]	PSO [36]	PSOPC [36]	HPSO [36]	DHPSACO [36]	MBA [37]	CBO [44]	ARO	I-ARO
A1	0.1960	7.22	4.49	4.9700	1.8000	0.1960	1.6200	1.8	1.8
A2	0.6020	1.80	1.457	1.2280	0.4420	0.5630	0.5630	0.563	0.442
A3	0.3070	1.13	0.111	0.1110	0.1410	0.4420	0.1110	0.111	0.111
A4	0.7660	0.196	0.111	0.1110	0.1110	0.6020	0.1110	0.111	0.111
A5	0.3910	3.09	2.620	2.8800	1.2280	0.4420	1.4570	1.457	1.266
A6	0.3910	0.785	1.130	1.4570	0.5630	0.4420	0.4420	0.563	0.563
A7	0.1410	0.563	0.196	0.1410	0.1110	0.1110	0.1110	0.111	0.111
A8	0.1110	0.785	0.111	0.1110	0.1110	0.1110	0.1110	0.111	0.111
A9	1.8000	3.09	1.266	1.5630	0.5630	1.2660	0.6020	0.442	0.563
A10	0.6020	1.228	1.457	1.2280	0.5630	0.5630	0.5630	0.442	0.563
A11	0.1410	0.111	0.111	0.1110	0.1110	0.1110	0.1110	0.111	0.111
A12	0.3070	0.563	0.111	0.1960	0.2500	0.1110	0.1110	0.111	0.111
A13	1.5630	1.990	0.442	0.3910	0.1960	1.8000	0.1960	0.196	0.196
A14	0.7660	1.620	1.457	1.4570	0.5630	0.6020	0.6020	0.563	0.563
A15	0.1410	1.563	1.228	0.7660	0.4420	0.1110	0.3910	0.442	0.442
A16	0.1110	1.266	1.457	1.5630	0.5630	0.1110	0.5630	0.563	0.602
Weight (lb)	427.203	1209.48	941.82	933.090	393.380	390.730	391.070	389.8142	389.4579
Worst weight (lb)	–	–	–	–	–	399.490	495.970	413.0482	417.4427
Mean weight (lb)	–	–	–	–	–	395.432	403.710	396.8514	396.5983
Standard deviation (lb)	–	–	–	–	–	3.0400	24.8000	5.3870	5.3506

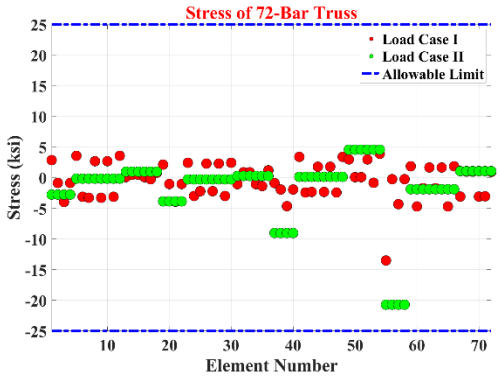
[DOI: 10.22068/ijoce.2024.14.3.593]

DHPSAC: Harmony Search Algorithm
CBO: Colliding bodies optimization
IMBA: Improved Mine Blast Algorithm

For the 72-bar truss design example, the displacement and stress design constraints for the best optimization run conducted by the I-ARO are provided in Fig. 20 in which the capability of the constraint handling approach in this paper is in perspective.



(a)



(b)

Figure 20. Displacement (a) and stress (b) design constraints for 72-bar truss problem

6.5. 160-bar Truss Structure

The progression of convergence for the ARO and I-ARO methods in handling the 160-bar truss design case is visualized in Fig. 21 which displays the superiority of the improved method over the standard one regarding the best optimization runs of both algorithms.

Table 6 shows the optimal design solution for the 160-bar truss problem obtained through 50 independent optimization runs using the I-ARO algorithm. The table also includes accompanying statistical outcomes and discrete design variables for the sake of comparison. The minimum achievable weight for the structure is computed using I-ARO, and outcomes from other established metaheuristic methods found in the literature are also included, enhancing the understanding of I-ARO's potential. It can be concluded that I-ARO effectively yields a weight of 1345.20 kg, representing the lowest feasible weight for this structure based on the reported results.

[DOI: 10.22068/ijoce.2024.14.3.593]

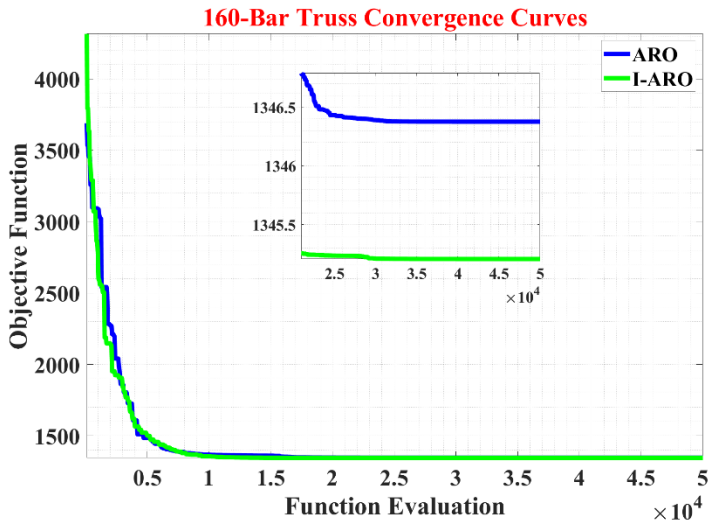


Figure 21. Convergence history of I-ARO and ARO for 160-bar truss structure

Table 6: Comparative results of ARO, I-ARO and other approaches in dealing with 160-bar truss problem

Design Variables (in.2)	SDR [45]	RGA [46]	RBAS [47]	DE [48]	AEDE [48]	ARO	I-ARO
A1	19.03	19.03	19.03	19.03	19.03	19.03	19.03
A2	5.27	5.27	5.27	5.27	5.27	5.27	5.27
A3	19.03	19.03	19.03	19.03	19.03	19.03	19.03
A4	5.27	5.27	5.27	5.27	5.27	5.27	5.27
A5	19.03	19.03	19.03	19.03	19.03	19.03	19.03
A6	5.75	5.75	5.75	5.75	5.75	5.75	5.75
A7	17.03	15.39	15.39	17.03	15.39	15.39	15.39
A8	6.25	5.75	5.75	5.75	5.75	5.75	5.75
A9	13.79	13.79	13.79	13.79	13.79	13.79	13.79
A10	6.25	5.75	5.75	5.75	5.75	5.75	5.75
A11	5.75	5.75	5.75	6.84	5.75	5.75	5.75
A12	12.21	13.79	12.21	12.21	12.21	13.79	12.21
A13	6.84	6.25	6.25	7.44	6.25	6.25	6.25
A14	5.75	5.75	5.75	5.75	5.75	5.75	5.75
A15	2.66	2.66	3.47	6.84	3.88	2.66	2.66
A16	7.44	7.44	7.44	8.66	7.44	7.44	7.44
A17	1.84	1.84	1.84	2.26	1.84	1.84	2.26
A18	8.66	8.66	9.40	12.21	8.66	8.66	8.66
A19	2.66	2.66	2.66	3.88	2.66	2.66	2.66
A20	3.07	3.07	3.47	3.88	3.07	3.07	3.07
A21	2.66	2.66	3.07	3.88	2.66	2.66	6.25
A22	8.06	8.06	8.06	8.66	8.06	8.06	8.06
A23	5.27	5.27	5.75	6.25	5.75	5.27	5.75
A24	7.44	6.25	6.25	7.44	6.25	7.44	6.25

[DOI: 10.22068/ijoce.2024.14.3.593]

A25	6.25	5.75	5.75	9.4	5.75	5.75	5.75
A26	1.84	1.84	2.26	4.79	2.26	1.84	2.26
A27	4.79	4.79	4.79	6.25	4.79	4.79	4.79
A28	2.66	2.66	3.07	4.79	2.66	2.66	2.66
A29	3.47	3.47	3.47	4.79	3.47	3.88	3.47
A30	1.84	1.84	1.84	1.84	1.84	1.84	1.84
A31	2.26	2.26	3.88	2.66	2.26	2.66	3.07
A32	3.88	3.88	3.88	3.88	3.88	3.88	3.88
A33	1.84	1.84	1.84	2.26	1.84	1.84	1.84
A34	1.84	1.84	2.26	2.66	1.84	1.84	1.84
A35	3.88	3.88	3.88	4.79	3.88	3.88	3.88
A36	1.84	1.84	2.66	2.26	1.84	1.84	1.84
A37	1.84	1.84	3.47	3.88	1.84	1.84	2.66
A38	3.88	3.88	3.88	4.79	3.88	3.88	3.88
Weight (kg)	1359.781	1337.442	1348.905	1448.306	1336.634	1346.3763	1345.2063
Worst weight (kg)	–	–	1401.6323	1743.596	1410.611	1582.5558	1573.9283
Mean weight (kg)	–	–	1367.5275	1617.346	1355.875	1392.6752	1398.6182
Standard deviation (kg)	–	–	–	81.930	18.805	53.9090	68.1814

RGA: Regional Genetic Algorithm
SDR: Selective Dynamic Rounding
RBAS: Rank-Based Ant Colony Algorithm

Fig. 22 showcases the stress-related design constraints for the most optimal optimization run achieved through the I-ARO algorithm. This presentation includes constraints for eight distinct load scenarios, providing a clear view of the constraint handling approach's effectiveness.

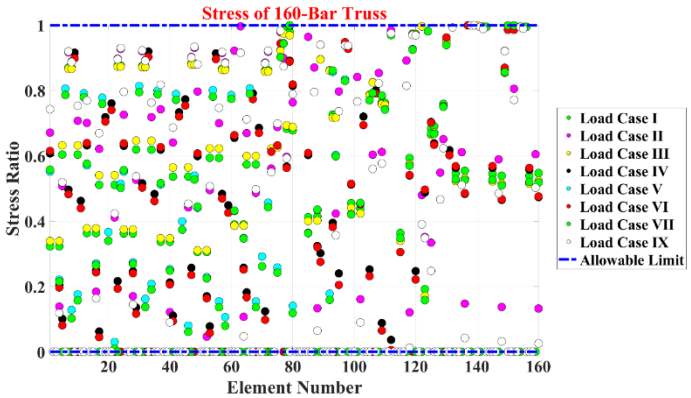


Figure 22. Stress design constraints for 160-bar truss problem.

[DOI: 10.22068/ijoce.2024.14.3.593]

For further studies, the presented method can be compared to those given in references [49-53].

7. CONCLUSIONS

This paper investigates the optimal design of truss structures with enhanced metaheuristic algorithms. For this purpose, the Improved Artificial Rabbits Optimization (I-ARO) algorithm is proposed for the first time in this paper in which the well-known random initialization process is substituted by the Diagonal Linear Uniform (DLU) initialization procedure. The key findings of this paper are as follows:

- Regarding the 10-bar truss problem, the lowest possible weight for the structure is calculated by the I-ARO algorithm as 5491.71 lb while the MBA with 5504.75 lb has the second rank.
- I-ARO can provide a best optimum value of 485.04 lb for the 25-bar truss structure while the ARO with 486.51 lb has the second rank.
- I-ARO algorithm is capable of reaching 1903.70 lb for the weight of the 52-bar truss structure which is better than the results of the ARO algorithm.
- For the 72-bar truss problem, I-ARO's superiority over ARO is obvious by providing 389.45 lb for the weight of this structure.
- I-ARO effectively yields a weight of 1345.20 kg for the 160-bar truss structure, representing the lowest feasible weight for this structure based on the reported results.

For future attempts, the capability of this structure can be tested by optimizing different types of engineering problems.

REFERENCES

1. Gomez AG, Oakes WC, Leone LL. Engineering your future: A project-based introduction to engineering. Great Lakes Press; 2004.
2. Lockhart SD, Johnson C. Engineering Design Communication: Conveying Design Through Graphics. Pearson College Div; 1999.
3. Martins JR, Ning A. Engineering design optimization. Cambridge University Press; 2021 Nov 18.
4. Rao SS. Engineering optimization: theory and practice. John Wiley & Sons; 2019 Nov 12.
5. Sonmez FO. Optimum design of composite structures: A literature survey (1969–2009). *J Reinf Plast* Compos 2017; **36**(1): 3-9.
6. Saka MP. Optimum design of steel frames using stochastic search techniques based on natural phenomena: a review. *Civ Eng Comput Tools Tech* 2007; **6**: 105-47.
7. Stolpe M. Truss optimization with discrete design variables: a critical review. *Struct Multidiscip Optim* 2016; **53**: 349-74.

[DOI: 10.22068/ijoce.2024.14.3.593]

8. Jiang F, Wang L, Bai L. An improved whale algorithm and its application in truss optimization. *J Bionic Eng* 2021; **18**: 721-32.
9. Tang H, Lee J. Chaotic enhanced teaching-based differential evolution algorithm applied to discrete truss optimization. In: *Structures* 2023 Mar 1; **49**: 730-747. Elsevier.
10. Astudillo NC, Hanchate DB, Jagtap AM. A Discrete Firefly Algorithm Applied to Structural Bridge Truss Optimization. In: International Conference on Smart Trends in Computing and Communications 2023 Jan 24; 301-332. Singapore: Springer Nature Singapore.
11. Altay O, Cetindemir O, Aydogdu I. Size optimization of planar truss systems using the modified salp swarm algorithm. *Eng Optim* 2023; 1-7.
12. Goodarzimehr V, Topal U, Das AK, Vo-Duy T. Bonobo optimizer algorithm for optimum design of truss structures with static constraints. In: *Structures* 2023 Apr 1; **50**: 400-417. Elsevier.
13. Mejías G, Zegard T. Simultaneous discrete and continuum multiresolution topology optimization. *Struct Multidiscip Optim* 2023; **66**(6): 137.
14. Mashru N, Patel P, Tejani GG, Kaneria A. Multi-objective Thermal Exchange Optimization for Truss Structure. In: Advanced Engineering Optimization Through Intelligent Techniques: Select Proceedings of AEOTIT 2022 2023 Apr 8; 139-146. Singapore: Springer Nature Singapore.
15. Shahrouzi M, Salehi A. An adaptive-band strategy to accelerate discrete optimization of space truss structures. *Int J Optim Civ Eng* 2023; **13**(2): 155-76.
16. Sellami M. A multi-stage descent algorithm for discrete and continuous optimization applied to truss structures optimal design. *Acta Mech* 2023; 1-21.
17. Sheng-Xue H. Truss optimization with frequency constraints using the medalist learning algorithm. In: *Structures* 2023 Sep 1; **55**: 1-15. Elsevier.
18. Khodadadi N, Çiftçioğlu AÖ, Mirjalili S, Nanni A. A comparison performance analysis of eight meta-heuristic algorithms for optimal design of truss structures with static constraints. *Decis Anal J* 2023; 100266.
19. Kale IR, Khedkar A, Sapre MS. Truss Structure Optimization Using Constrained Version of Variations of Cohort Intelligence. In: Optimization Methods for Structural Engineering 2023 Jun 7; 67-78. Singapore: Springer Nature Singapore.
20. Chen A, Lin X, Zhao ZL, Xie YM. Layout optimization of steel reinforcement in concrete structure using a truss-continuum model. *Front Struct Civ Eng* 2023; 1-7.
21. Ranjbarzadeh R, Razavi S, Tavakoli S. A comparative study of recent multi-objective metaheuristics for solving constrained truss optimisation problems. *Arch Comput Methods Eng* 2021; 1-17.
22. Seyedzadeh M, Asadian E. Comparative performance of twelve metaheuristics for wind farm layout optimisation. *Arch Comput Methods Eng* 2022; 1-14.
23. Saba F, Asadian E, Davoodi M. Optimum design of automobile components using lattice structures for additive manufacturing. *Mater Test* 2020; **62**(6): 633-639.
24. Amini K, Ghomi SF, Ayatollahi A. Robust design of electric vehicle components using a new hybrid salp swarm algorithm and radial basis function-based approach. *Int J Veh Des* 2020; **83**(1): 38-53.
25. Li J, Zhang H. A novel chaotic Henry gas solubility optimization algorithm for solving real-world engineering problems. *Eng Comput* 2020; **38**(2): 871-883.

[DOI: 10.22068/ijoce.2024.14.3.593]

26. Heidari AA, Heidari AA, Hossein Pour MR. A new chaotic Lévy flight distribution optimization algorithm for solving constrained engineering problems. *Expert Syst* 2022; e12992.
27. Shariat Panahi MH, Asadian E. Hunger games search algorithm for global optimization of engineering design problems. *Mater Test* 2022; **64**(4): 524-532.
28. Gharibzadeh S, Asadian E. African vultures optimization algorithm for optimization of shell and tube heat exchangers. *Mater Test* 2022; **64**(8): 1234-1241.
29. Asadian E, Shariat Panahi MH. Artificial gorilla troops algorithm for the optimization of a fine plate heat exchanger. *Mater Test* 2022; **64**(9): 1325-1331.
30. Wang L, Zhang H. Cheetah optimization algorithm for optimum design of heat exchangers. *Mater Test* 2023; **65**(8): 1230-1236.
31. Wang L, Cao Q, Zhang Z, Mirjalili S, Zhao W. Artificial rabbits optimization: A new bio-inspired meta-heuristic algorithm for solving engineering optimization problems. *Eng Appl Artif Intell* 2022; **114**: 105082.
32. Li Q, Bai Y, Gao W. Improved initialization method for metaheuristic algorithms: A novel search space view. *IEEE Access* 2021; **9**: 121366-84.
33. Awadallah MA, Braik MS, Al-Betar MA, Abu Doush I. An enhanced binary artificial rabbits optimization for feature selection in medical diagnosis. *Neural Comput Appl* 2023; 1-56.
34. Wang Y, Huang L, Zhong J, Hu G. LARO: Opposition-based learning boosted artificial rabbits-inspired optimization algorithm with Lévy flight. *Symmetry* 2022; **14**(11): 2282.
35. Rajeev S, Krishnamoorthy CS. Discrete optimization of structures using genetic algorithms. *J Struct Eng* 1992; **118**(5): 1233-50.
36. Li LJ, Huang ZB, Liu F. A heuristic particle swarm optimization method for truss structures with discrete variables. *Comput Struct* 2009; **87**: 435-43.
37. Sadollah A, Bahreininejad A, Eskandar H, Hamdi M. Mine blast algorithm for optimization of truss structures with discrete variables. *Comput Struct* 2012; 102-103: 49-63.
38. Wu SJ, Chow PT. Steady-state genetic algorithms for discrete optimization of trusses. *Comput Struct* 1995; **56**: 979-91.
39. Park HS, Sung CW. Optimization of steel structures using distributed simulated annealing algorithm on a cluster of personal computers. *Comput Struct* 2002; **80**(14-15): 1305-16.
40. Erbatur F, Hasançebi O, Tütüncü İ, Kılıç H. Optimal design of planar and space structures with genetic algorithms. *Comput Struct* 2000; **75**(2): 209-24.
41. Ming-Zhu D. An improved Templeman's algorithm for the optimum design of trusses with discrete member sizes. *Eng Optim* 1986; **9**(4): 303-12.
42. Lee KS, Geem ZW, Lee S, Bae K. The harmony search heuristic algorithm for discrete structural optimization. *Eng Optim* 2005; **37:** 663-84.
43. Kaveh A, Talatahari S. A particle swarm ant colony optimization for truss structures with discrete variables. *J Constr Steel Res* 2009; **65**(8-9): 1558-68.
44. Kaveh A, Talatahari S. Hybrid charged system search and particle swarm optimization for engineering design problems. *Eng Comput* 2011; **28**(4): 423-440.
45. Groenwold AA, Stander N. Optimal discrete sizing of truss structures subject to buckling constraints. *Struct Optim* 1997; **14**: 71-80.

[DOI: 10.22068/ijoce.2024.14.3.593]

46. Groenwold AA, Stander N, Snyman JA. A regional genetic algorithm for the discrete optimal design of truss structures. *Int J Numer Methods Eng* 1999; **44**: 749-66.
47. Capriles PVSZ, Fonseca LG, Barbosa HJC, Lemonge ACC. Rank-based ant colony algorithms for truss weight minimization with discrete variables. *Commun Numer Methods Eng* 2007; **23**: 553-75.
48. Ho-Huu V, Nguyen-Thoi T, Vo-Duy T, Nguyen-Trang T. An adaptive elitist differential evolution for optimization of truss structures with discrete design variables. *Comput Struct* 2016; **165**: 59-75.
49. Kaveh A. Advances in Metaheuristic Algorithms for Optimal Design of Structures. 3rd ed. Springer International Publishing; 2021.
50. Kaveh A. Applications of Metaheuristic Optimization Algorithms in Civil Engineering. Springer; 2017.
51. Kaveh A, Bakhshpoori T. Metaheuristics: Outlines, MATLAB Codes and Examples. Springer; 2019.
52. Kaveh A, Mahdavi VR. Colliding Bodies Optimization: Extensions and Applications. Springer; 2015.
53. Kaveh A, Dadras Eslamlou A. Metaheuristic Optimization Algorithms in Civil Engineering: New Applications. Springer; 2020.

[DOI: 10.22068/ijoce.2024.14.3.593]