Scientia Iranica A (2022) 29(6), 2995–3015

SCIENTIA IRANICA

Sharif University of Technology
Scientia Iranica
Transactions A: Civil Engineering
http://scientiairanica.sharif.edu

Optimum structural design of spatial truss structures via migration-based imperialist competitive algorithm

A. Kaveh* **and F. Rajabi**

School of Civil Engineering, Iran University of Science and Technology, Narmak, Tehran-16, Iran.

Received 3 November 2021; received in revised form 15 December 2021; accepted 3 January 2022

KEYWORDS
Hybrid algorithm;
Imperialist competitive algorithm;
Biogeography-based optimization;
Metaheuristic algorithms;
Optimum design;
Truss structures design;
Structural optimization.

Abstract. The current study presents a new hybrid algorithm generated by combining advantageous features of Imperialist Competitive Algorithm (ICA) and Biogeography-Based Optimization (BBO) to establish an effective search technique. Although the ICA performs fairly well at the exploration phase, it is less effective at the exploitation stage. In addition, its convergence speed is problematic in some instances. Meanwhile, the migration operator of BBO method strongly emphasizes the local search to find the optimum solution more precisely. The combination of these two algorithms generates a robust hybrid algorithm that enjoys both exploratory and exploitative functionalities. The proposed hybrid algorithm is called Migration-Based Imperialist Competitive Algorithm (MBICA). To validate its performance, MBICA is used to optimize a variety of benchmark truss structures. Compared to some other methods, this algorithm converges to better or at least identical solutions by reducing the required number of structural analyses. Finally, the results from the standard BBO, ICA, and other recently developed metaheuristic optimization methods were compared with those obtained in this study.

1. Introduction

The cost of materials has always been a significant factor in constructing structures, and minimizing the total weight of a structure has been considered a way to reduce the overall construction costs. While determining the optimal design of structures, the primary objective is to minimize the weight of the structures within the constraints of their design code and specification. Gradient-based methods and metaheuristics are two general categories of optimization algorithms which are widely developed for the research and industrial purposes such as structural optimization. Although the first group ensures optimality of the obtained solutions, they are sometimes time consuming and dependent on primary necessities and gradient information [1]. Since structural optimization considers the properties of the structural elements as design variables, effective application of such methods is impossible due to the relatively large number of structural elements. On the contrary, metaheuristics provide an approximate solution within a reasonable time. In addition, their applicability to almost all disciplines makes them popular among practitioners and researchers [2]. Such methods ease the process of finding a global optimum solution by mimicking a simple or complex strategy with no requisite for derivative information of the problem. Researchers benefit from the advantages and robust tools they provide for optimal design of large-scale structures with complex geometry and thousands of components. Inspired by nature, three types of metaheuristic algorithms are proposed, the first of which is Evolutionary Algorithms (EAs). Repro-

*. *Corresponding author. Tel.: +98 21 44202710;*
Fax: +98 21 77240398
E-mail addresses: alikaveh@iust.ac.ir (A. Kaveh);
farivar.rajabi@gmail.com (F. Rajabi)

doi: 10.24200/sci.2022.59344.6188

duction, mutation, recombination, and selection are among the evolutionary mechanisms included in these algorithms. Genetic Algorithm (GA) developed by Holland [3] based on Darwin's theories and Differential Evolution (DE) [4] are other popular algorithms of this category performed based on evolutionary principles. The second group of algorithms includes the algorithms of swarm intelligence that function based on the social and biological behavior of the creature groups. For example, inspiration from the movement of flocks of birds and fish schools has triggered the development of Particle Swarm Optimization (PSO) [5]. Ant Colony Optimization (ACO) [6] is based on food finding mechanisms of ants that use pheromones to communicate. The third group of algorithms simulates natural physical and chemical laws, including gravity, annealing, and collision. The Big Bang-Big Crunch (BB-BC) [7], Simulated Annealing (SA) [8], Charged System Search (CSS) [9], and Colliding Bodies Optimization (CBO) [10] are well-known cases in point. Numerous studies in the literature have been conducted on the application of different versions of the above-mentioned algorithms for the optimal design of the practical systems and models with complex configurations [11–13]. In order to minimize the weight of different types of structures in structural optimization to achieve an optimal design under specific criteria, Kaveh et al. [14] proposed Enhanced Forensic-Based Investigation (EFBI) for dome truss optimization problems with frequency constraints. Azad et al. [15] developed a computationally efficient multi-stage guided stochastic search algorithm for optimization and standardization of real-size free-form steel double-layer grids. Kaveh et al. [16] represented a new version of CBO called New Enhanced Colliding Bodies Optimization (NECBO) to optimize the steel portal frames. The monitored convergence curve method is introduced and validated to improve the performance of metaheuristic structural optimization algorithms [17]. Moreover, some strategies such as use of the chaotic maps embedded in metaheuristics [18] and Upper Bound Strategy (UBS) [19] have been investigated to date in order to enhance the robustness of the optimization algorithms and reduce the computational time, respectively.

In metaheuristic algorithms, the two pivotal factors that considerably affect their functionality are exploration (diversification) and exploitation (intensification). Exploitation serves a paramount role in searching the space close to solutions with more fitness. In contrast, exploration enables the algorithm to focus on the unexploited search space areas, often through randomization. Trapped in the local optima and failed to reach promising solutions with a low convergence rate are the consequences of deficiency in the exploration and exploitation, respectively. Therefore, a proper counterbalance between exploitation and exploration yields promising results. However, the main drawbacks of some existing metaheuristic algorithms include premature convergence and weak exploitation capability, resulting in being trapped in the local optima rather than finding global optimum and slow convergence, respectively. Recent studies have addressed such issues by improving standard algorithms or hybridizing different algorithms by merging their beneficial specifications [20]. Hybridization is the process of mixing two or more algorithms to achieve better outcomes by taking advantage of the properties of different algorithms. Several attempts have been made to combine different algorithms to enhance their performance in the optimum design of structures. Particle Swarm Optimizer Cultural (PSOC) algorithm was introduced by Jafari et al. [21] for the optimal design of truss structures. Electromagnetism-like Firefly Algorithm (EFA) was developed by Le et al. [22] for discrete structural optimization. In order to optimize the shape and size of truss structures under multiple frequency constraints, Adaptive Hybrid Evolutionary Firefly Algorithm (AHEFA) was introduced by Lieu et al. [23]. Hybrid Butterfly Optimization Algorithm and Symbiosis Organism Search (h-BOASOS) [24] for weight and cost optimization of the cantilever retaining walls, hybrid Eagle Strategy with Differential Evolution (ES-DE) [25] for optimum design of frame structures, and hybrid Multi-Design Variable Configuration cascade optimization and Vibrating Particles System (MDVC-UVPS) [26] for large-scale dome truss structures optimization have been some of other noteworthy hybrid methods in structural design field in recent years.

The current study proposes a hybrid algorithm that embeds Biogeography-Based Optimization (BBO) [27] into Imperialist Competitive Algorithm (ICA) [28] to design optimum trusses with discrete and continuous variables. Several truss optimization problems are used to determine the effectiveness of the developed algorithm. The results of the standard BBO, ICA, and different other methods reported in the literature were compared with those of the proposed method. The present study is composed of five sections. Followed by an introduction in Section 1, a description of each BBO, ICA, and hybrid algorithm is outlined in Section 2. Five benchmark truss structures are discussed in Section 3 that demonstrate the algorithm function in the constrained structural engineering problems. The efficiency of the Migration-Based Imperialist Competitive Algorithm (MBICA) is investigated in Section 4. The concluding remarks are given in Section 5.

2. Optimization algorithms

Before the MBICA proposal for truss design optimization, the BBO and ICA general steps are presented in the next section to make the research project self-explanatory. In addition, some explanation on how to combine these two algorithms is given to ensure enhanced search results.

2.1. Biogeography-Based Optimization (BBO)

The BBO is a population-based metaheuristic search method similar to GA, ACO, and PSO. Developed by Simon [27], BBO is categorized as the EAs taking its inspiration from the biogeography. Biogeography investigates how organisms are geographically distributed throughout nature. The species migration from one island to another, appearance of new species, and extinction are mathematically described in the probability-based biogeographic models. An island is an area of suitable habitat that is geographically separated from other habitats. The living condition of each habitat is determined by Habitat Suitability Index (HSI). An island has a high HSI if it is well suited to living with biological species. Variables that describe the characteristics of habitat are termed Suitability Index Variables (SIVs). They are habitat-independent variables that can be used to calculate HSI for each habitat. Emigration describes the process by which some species move from one habitat to another. The process of a species entering one habitat from an external habitat is recognized as immigration. Habitats with a high HSI are generally capable of sustaining many species while those with a low HSI are mainly composed of a relatively fewer species. Generally, the more flocked a habitat is, the more static its species distribution will be. Therefore, habitats with a high HSI are likely to take a small value as the species immigration rates λ and high value as the emigration rates μ since they are almost saturated with species. In contrast, low HSI habitats with sparse population take a greater species immigration rate. In the BBO algorithm, each candidate solution considered as a habitat (H_i) has some decision variables denoted by SIVs while determining the position of solution in the search space. HSI for a candidate solution representing its quality is determined based on the fitness function value. Mathematically, solutions with low fitness values have higher immigration rates; therefore, they are more likely to borrow features from other candidates, leading to improvements in the next algorithm cycles. The immigration rate of better solutions is lower than that of low-fitness ones, meaning that they are less affected by other ones. The emigration rate also works similarly. Through two main mechanisms of migration and mutation, the BBO algorithm moves the population towards better solutions. The major steps of BBO are elaborated in the following:

Step 1. Definition of problem and parameters: First, the optimization problem, size of population (nH), number of maximum function evaluations ($\max NFEs$), number of design variables ($nVar$), and design variables boundaries ($Lb.Ub$) are defined.

Step 2. Initialization: A random population is generated based on the population size, design variables number, and design variables limits. The candidate solutions can be considered as nH habitats. The set of habitats (H) can be generated through the following equation:

$$S = Lb + (Ub - Lb) \times rand(nH.nVar). \tag{1}$$

Step 3. Migration: The HIS value of each habitat is first determined by evaluating the penalized objective function ($PFit$). Then, emigration rate μ_i and immigration rate λ_i for all habitats are computed as follows:

$$\lambda_i = I\left(1 - \frac{K}{S_{\max}}\right), \tag{2}$$

$$\mu_i = E\left(\frac{K}{S_{\max}}\right), \tag{3}$$

where I and E are the maximum possible immigration and emigration rates, respectively; K is the species number of the ith habitat determined according to habitat HSI; and $S_{\max}$ maximum number of the species. λ_i and μ_i control the process of information sharing between the habitats. Different migration models are useable for computing these rates. Figure 1 shows a case where $E = I$ as a simple linear migration model, and both of these rates are set at 1. Based on Figure 1, it is evident that habitats with a high HSI (good candidates), such as S_2, have a low rate of immigration (λ_i) and high rate of emigration (μ_i). Hence, the better habitats are more likely to share their valuable information with other ones. Moreover, the habitat with low HSI (poor candidates), such as S_1, has a low rate of emigration (μ_i) and a high rate of immigration (λ_i). In other words, these habitats alter their positions

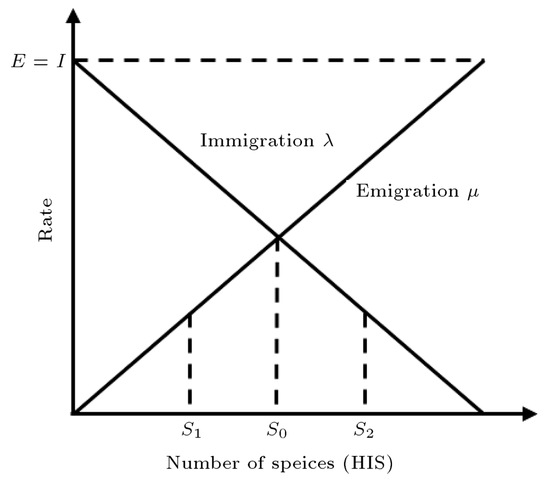


Figure 1. Simple linear migration model.

```
Inputs: algorithm and problem parameters
Outputs: The best solution
Generate the random population of habitats X_i (i = 1:nH)
Evaluate HSI for each habitat
While (stopping condition is not met) do
    Calculate the immigration rate λ, and the emigration rate μ using
    simple linear migration model, for each habitat
    For i = 1 to nH
        For k = 1 to D
            Habitat migration:
            If rand < λ_i Then
                Select X_j with probability based on emigration rate μ_i
                If X_j is selected Then
                    Select kth variable of emigrating habitat X_j
                    Replace selected variable in the immigrating habitat X_i
                End
            End
            Habitat mutation:
            Select kth variable of X_i with probability m_i
            If X_ik is selected Then
                Replace X_ik with randomly generated SIV
            End
        End
    End
End
Return X_min
```

Algorithm 1. The Biogeography-Based Optimization (BBO) pseudocode.

by taking information from other habitats with a higher possibility while transferring their information with different habitats at a relatively low possibility. Further, S_0 refers to the point of equilibrium where λ_i equals μ_i. Of note, S_0 has the same chance of sharing information with other habitats or receiving information from them. After computing μ_i and λ_i, the migration process for all habitats becomes as follows:

$$H_i(SIV) \leftarrow H_j(SIV), \tag{4}$$

where H_i and H_j represent the immigrating and emigrating habitats, respectively. For each immigrating habitat, an emigrating habitat is selected by applying a selection methodology like roulette wheel selection. Emigration rate (μ_i) determines the probability of choosing H_j as an emigrating habitat. Each variable of H_i is replaced with λ_i probability by the corresponding H_j variables.

Step 4. Mutation: Following the migration phase, the mutation operator modifies the position of the habitats. Based on its mutation rate (m_i), the mutation operator randomly alters a habitat's SIV by choosing a value from the feasible search domain. The rate of mutation for habitats is calculated as follows:

$$m_i = m_{\max}\left(\frac{1-P_i}{P_{\max}}\right), \tag{5}$$

where $m_{\max}$ is the user that defines a control parameter $P_{\max} = \max\{P_i\}$, P_i the associated probability of the ith habitat, and $P_{\max}$ the maximum habitat probability in the population.

Step 5. Terminal condition check: If the algorithm termination conditions are met, the algorithm ends. If not, go to the third step.

The BBO pseudocode is presented in Algorithm 1.

2.2. Imperialist Competitive Algorithm (ICA)

The ICA was generated by simulating the social-political procedure of imperialistic competition. Similar to other EAs, this algorithm begins with a number of individuals referred to as a country that is either an imperialist or a colony. A number of colonies with an imperialist together establish an empire. There are a defined number of empires at the beginning of the searching process within the search space. A particular number of the best individuals are chosen as to be the imperialists. Through a competition, imperialists take a number of countries as colonies. Throughout generations, weak imperialists collapse and powerful imperialists compete with each other in order to take control of their colonies until one empire finally survives. ICA consists of two main mechanisms, i.e., improving the colonies owned by each imperialist through direct interaction with its imperialist and taking possession of colonies from other empires. As a result of the first one, empires themselves are empowered. Upon doing so, every colony has the chance to fulfill the role of the emperor of an empire. Throughout the second mechanism with the competition between empires, the weakest empire loses its weakest colony and the strongest empire takes possession of that.

This process will continue until the weakest empire breaks down. The emperor's quality and its colonies as well as the number of colonies determine the power of each imperialist or empire. Through the colonial improvement and imperialistic competition, the population is led to more promising areas of search space. The ICA main body repeats the mentioned methods to meet and satisfy the stopping criterion. The ICA consists of four stages: formation of the primary empires, colony evolution, imperialist revision, and imperialist competition. A summary of the key steps of the algorithm is presented below [29]:

Step 1. Problem and parameters definition: First, the optimization problem, size of population (nC), function evaluations maximum number ($\max NFEs$), design variables number ($nVar$), and design variables limits ($Lb.Ub$) are first defined. In addition, the algorithm parameters such as the number of empires and assimilation factors are selected.

Step 2. Initialization: Select a set of random initial solutions on the search space. Each candidate can be considered as a country. The group of countries (C) is obtained based on the following equation:

$$C = Lb + (Ub - Lb) \times rand(nC.nVar). \tag{6}$$

Step 3. Formation of the initial empires: First, the penalized objective function ($PFit$) for each country is calculated. Then, the specific number ($nImp$) of the strongest countries (the countries whose $PFit$ value is superior) will be chosen to be imperialists. Meanwhile, the rest of the countries ($nC - nImp$) will become colonies. The number of colonies taken by an imperialist must correlate straightly to its power. A Normalized Cost (NC) for an emperor is calculated to divide the colonies proportionately among imperialists, as shown in the following:

$$NC_{Imp_i} = PFit_{Imp_i} - \max(PFit_{Imp})$$

$$i - 1, ..., nImp, \tag{7}$$

where the max function returns the maximum penalized objective function value of the imperialists as the worst. Normalized Power (NP) is determined for each imperialist as follows:

$$NP_{Imp_i} = \left| \frac{NC_{Imp_i}}{sum(NC)} \right|, \tag{8}$$

where the *sum* function computes the total NCs of all imperialists. NP is the portion of colonies belonging to the ith imperialist. Initially, the following number of colonies ($nCol$) of each imperialist is obtained below:

$$nCol_i = round\,(NP_{Imp_i} \times (nC - nImp))$$

$$i = 1, ..., nImp, \tag{9}$$

where *round* stands for the rounding function.

Step 4. Colonies movement: The assimilation policy is pursued by the imperialist states to improve the colonies. This method is simulated by relocating all the colonies into their imperialist. In case a colony occupies a more promising place in the solution space, it may be replaced by its imperialist. The jth colony movement from the ith empire ($Col_{i.j}$) occurs in a uniform random direction toward its imperialist (Imp_i) and a uniform random deviation as follows:

$$direction = \beta \times U\,(0.(Col_{i.j} - Imp_i)) \tag{10}$$

$$i = 1, ..., nImp \;\text{ and }\; j = 1, ..., nCol_i,$$

$$Deviation = U(-\gamma. + \gamma), \tag{11}$$

$$newCol = Col + direction \times deviation, \tag{12}$$

where *newCol* determines the position of the new colonies as a uniform random number which is distributed between a and b. β created by the function of $U(a.b)$; β that is greater than one is another parameter. The parameter γ determines the deflection from the original direction. In most cases, good algorithm convergence occurs when the value is about 2 for β and about $\pi/4$ (Rad) for γ.

Step 5. Revolution: Application of the revolution operator to some countries.

Step 6. Updating of the imperialist: If the new position of the colony has a better-penalized objective function than that of its corresponding imperialist, the imperialist and colony exchange their positions.

Step 7. Empire total power: The total power of an empire influenced by the related imperialist state power and colonies power is calculated as follows:

$$TC_i = PFit_{Imp_i} + \xi \times mean(PFit_{Col_i})$$

$$i = 1, ..., nImp, \tag{13}$$

where TC_i is the ith empire total power and ξ parameter is chosen between 0 and 1.

Step 8. Imperialist competition: Each empire aims to control other imperialist colonies. With imperialist competition, weaker empires gradually become weaker, while stronger ones become more powerful. This process is modeled by choosing one of the

```
Inputs: algorithm and problem parameters
Outputs: The best solution
Generate the population randomly X_i (i = 1:nC)
Initialize the empires:
For (each country) do
        Sort the countries according to penalized fitness function
        Select nImp (number of imperialist countries) out of nC
        Calculate each imperialist normalized cost and normalized power
        Assign rest nCol countries to selected imperialists
End
While (stopping condition is not met) do
        For j=1 to nImp do
                Assimilation:
                Move the colony toward their imperialist
                Compute the cost of assimilated colonies
                Revolution:
                Revolve the new colony with a specific probability
                Update Imperialist
                If the new colony has a lower cost than the imperialist, Then
                        Exchange the position of colony and imperialist
                End
        End
        Imperialist competitive:
        Pick the weakest colony from the weakest empire and assign it to the
        empire that has the most chance to take it
        Elimination process:
        If there is an imperialist with no colonies, Then
                Eliminate the imperialist
        End
End
Return X_min
```

Algorithm 2. Imperialist Competitive Algorithm (ICA) pseudocode.

weakest colonies of the poorest empire and launching competition among all the imperialists to take over this colony. There is more opportunity for more powerful imperialist countries to take possession of the picked colony. First, given the total power of each empire to start the competition, possession possibility of each empire is calculated. Then, the total NC in each empire is defined as follows:

$$NTC_i = TC_i - \max(TC) \qquad i = 1, ..., nImp, \quad (14)$$

where NTC_i is the ith empire normalized total cost. According to the normalized total cost, each empire has the following possession probability:

$$P_i = \frac{NTC_i}{sum(NTC)} \qquad i = 1, ..., nImp, \quad (15)$$

where the sum function determines the normalized total costs summation of all empires.

To create a possession index vector (D), a uniformly random vector with the same size as that of the possession vector (P) is generated from [0.1] interval and subtracted from it, as shown below:

$$D_i = P_i - rand(0.1) \qquad i = 1, ..., nImp. \quad (16)$$

The control of the weakest colony of the weakest empire will be taken by the empire with the maximum relevant index in D.

Step 9. Eliminating the weakest empire: In case an empire misses all its colonies, it is assumed to collapse. In this respect, the imperialist will also be owned by the empire with the maximum value of D.

Step 10. Terminal condition check: If the algorithm termination conditions are met, the algorithm ends. Otherwise, go to Step 4.

The ICA pseudocode is presented in Algorithm 2.

2.3. The Migration-Based Imperialist Competitive Algorithm (MBICA)

Upon combining the exclusive features of both BBO and ICA techniques, a structural optimization algorithm called MBICA can be developed. This algorithm is classified as a sequential high-level relay type and a heterogeneous hybrid algorithm with a global search [30]. During the optimization process, the BBO algorithm employs the migration operator to exchange information among habitats which can be regarded as a robust mechanism for exploitation. However, the investigations show that this algorithm is sometimes prone to premature convergence and state of being trapped in the local minima due to its relatively weak exploration ability and insufficient population diversity. On the contrary, ICA has an impressive ability to explore the search space. In the early iterations stage, the colonies of each empire spread out over the entire search space and move in an enormous area of search space in the assimilation process.

Accordingly, due to the competition among multiple empires, colonies can be transferred from one empire to another. In this way, diversity of population can be ensured. To develop a new population-based metaheuristic with appropriate exploration and exploitation capabilities, a hybrid method called MBICA is introduced. Upon using the multi-population structure of the ICA, the performance of the new algorithm in the search mechanism is enhanced. Based on this technique, each subpopulation can independently modify the solutions in separated groups. In this way, instead of focusing on one area, the solutions are distributed across the entire search space. As a result, the algorithm can explore different search space areas simultaneously since the subpopulations can be located in different locations within the search space. On the contrary, BBO encourages the poor solutions to use the information of the good ones. As a result of this strong local search mechanism, the population can concentrate on successful candidate solutions to determine the optimum solution with greater accuracy, thus providing a greater convergence rate for the new optimization method. The major algorithm steps are as follows:

Step 1. Problem and parameters definition: First, define the optimization problem, population size (nP), function evaluations maximum number ($\max NFEs$), design variables number ($nVar$), and design variables limits ($Lb.Ub$). Then, the algorithm parameters such as the number of empires (subpopulations) are selected.

Step 2. Initialization: Generate a certain number of feasible solutions randomly based on a limited space.

Step 3. Formation of initial empires (subpopulations): Similar to the standard ICA, establish initial empires.

Step 4. Immigration and emigration rates: Estimate the immigration λ_i and emigration μ_i rates of each candidate solution within the population based on its fitness. As previously proposed, based on the simplified linear model for migration, the sum of λ_i and μ_i for each country should be equal to unity.

Step 5. Migration: Modify the colonies of each empire according to the basic BBO algorithm. This phase involves implementing the migration operator separately for each subpopulation.

Step 6. Assimilation: Move each empire colony toward its corresponding imperialist.

Step 7. Revolution: Revolve some countries.

Step 8. Imperialist updating: If the new position of a colony has a penalized objective function superior to its related imperialist, the imperialist and colony exchange their positions.

Step 9. Empire total power: Compute the total power of an empire, considering the instructions given in the basic ICA.

Step 10. Imperialist competition: Select the poorest colony of the weakest subpopulation. All imperialists compete to possess this colony. The more powerful imperialists have a greater chance of taking possession of the picked colony.

Step 11. Eliminating the weakest empire: Whenever a subpopulation misses all its colonies, consider it eliminated. In this regard, the imperialist itself will also be owned by the imperialist with the maximum corresponding D value.

Step 12. Terminal condition check: If the algorithm termination conditions are met, the algorithm ends. If not, go to Step 4.

The hybrid algorithm flowchart is demonstrated in Figure 2.

3. Design of truss structures

To show the efficiency of the developed optimization method in this section, five truss structures, which are the size optimization benchmarks in structural engineering, with discrete and continuous variables were studied. In order to optimize the size of structures, efforts were made to minimize the cross-sectional elements that guaranteed accepted construction costs. It also eliminates some limitations that restrict design variables and structural responses. The optimal design problem is presented below:

$$\text{Min } W(\{X\}) = \sum_{i=1}^{nm} \gamma_i.L_i.A_i(x_i),$$

s.t.:

$$\delta_{\min} \le \delta_i \le \delta_{\max} \qquad i = 1, ..., nm, \tag{17}$$

$$\sigma_{\min} \le \sigma_i \le \sigma_{\max} \qquad i = 1, ..., nm,$$

$$\sigma_i^b \le \sigma_i \le 0 \qquad i = 1, ..., ns,$$

$$A_{\min} \le A_i \le A_{\max} \qquad i = 1, ..., nm,$$

where $W(\{X\})$ indicates the structure weight, $\{X\}$ the vector containing design variables, nm the number of

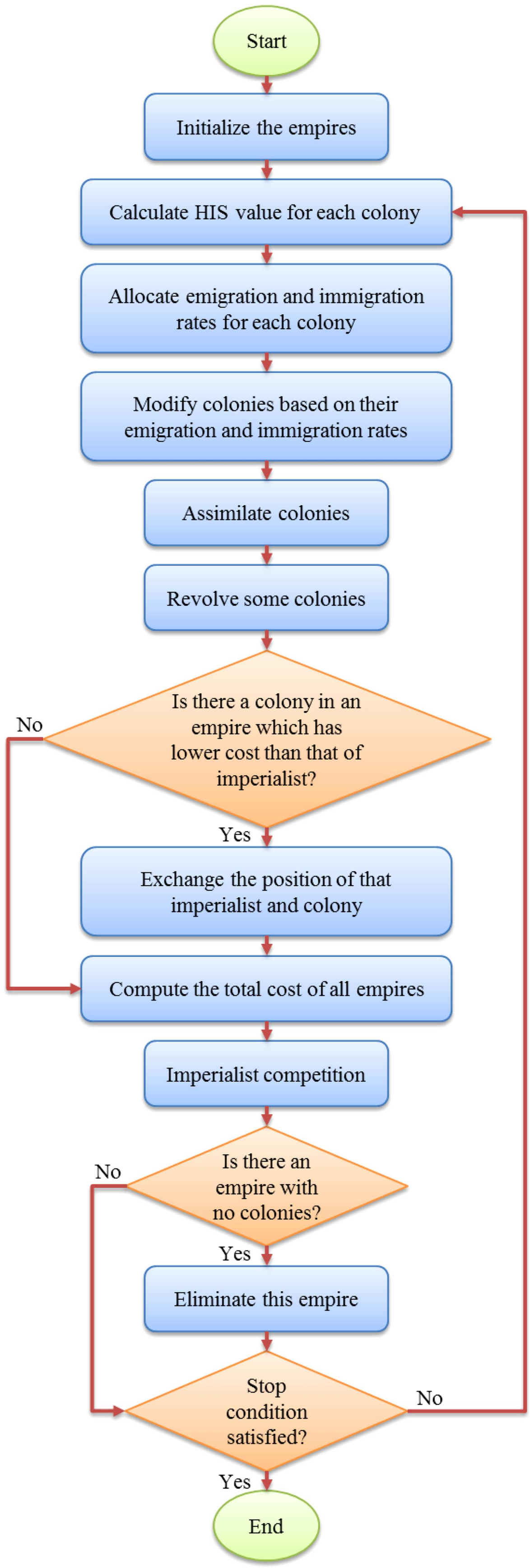


Figure 2. Flowchart of the hybrid algorithm.

structure members. In addition, γ_i and L_i stand for the ith member material density and length, respectively. Moreover, δ_i is the displacement of node i, σ_i the stress of the ith element, and σ_i^b the permissible buckling load when the ith member is subjected to compression. The number of compression elements is denoted by ns. The cross-sectional area for the truss member of the ith is A_i. The upper and lower bounds are indicated by the max and min subtitles, respectively. In discrete optimization, sections are selected from a predefined list using the same formula. The penalty approach is used to take into account the limitations. Here, the objective function is defined as follows:

$$f(x) = (1 + \varepsilon_1 . v)^{\varepsilon_2} \times W(\{X\}),$$

$$v = \sum_{i=1}^{nc} \max\left(0 . g_j(\{X\})\right), \qquad (18)$$

where the sum of the violations of the design constraints is denoted by v. The constant setting ε_1 is equal to 1 while ε_2 starts at 1.5 and increases linearly to 3 at the end.

In the following subsections, upon using the mentioned formulations through 30 independent calculations, the algorithm results were measured and compared in terms of the minimum, average, and standard deviations to provide a statistically meaningful comparison. The internal parameters of the BBO and ICA were then adjusted according to the literature [31,32]. Parameters of the MBICA for 25-bar spatial truss, 72-bar space truss, 120-bar dome truss, and 272-bar transmission tower were set as follows: population size = 20, number of subpopulations = 8, and revolution probability = 0.1. The initial population, number of subpopulations, and revolution probability of the MBICA for 582-bar tower truss were assumed to be 30, 3, and 0.2, respectively. More detailed geometric properties and loading conditions for each structure can be found in [2].

3.1. 25-bar spatial truss

The twenty-five-bar space truss represented in Figure 3 is frequently used in the literature as a design example to compare different optimization algorithms. The structural elements are categorized into eight groups. Table 1 lists the structure design variables and allowable stress values for all groups. For the variables, the upper and lower bounds are 0.1 and 3.4 in^2, respectively. In all nodes, the maximum displacement limitations of $\pm$0.35 in are considered in each direction. The elasticity modulus and truss density of the members are 10,000 ksi and 0.1 lb/in^3, respectively. According to Table 2, two loading modes are applied on this truss.

Table 3 makes a comparison between the optimal designs obtained by the MBICA, standard algorithms

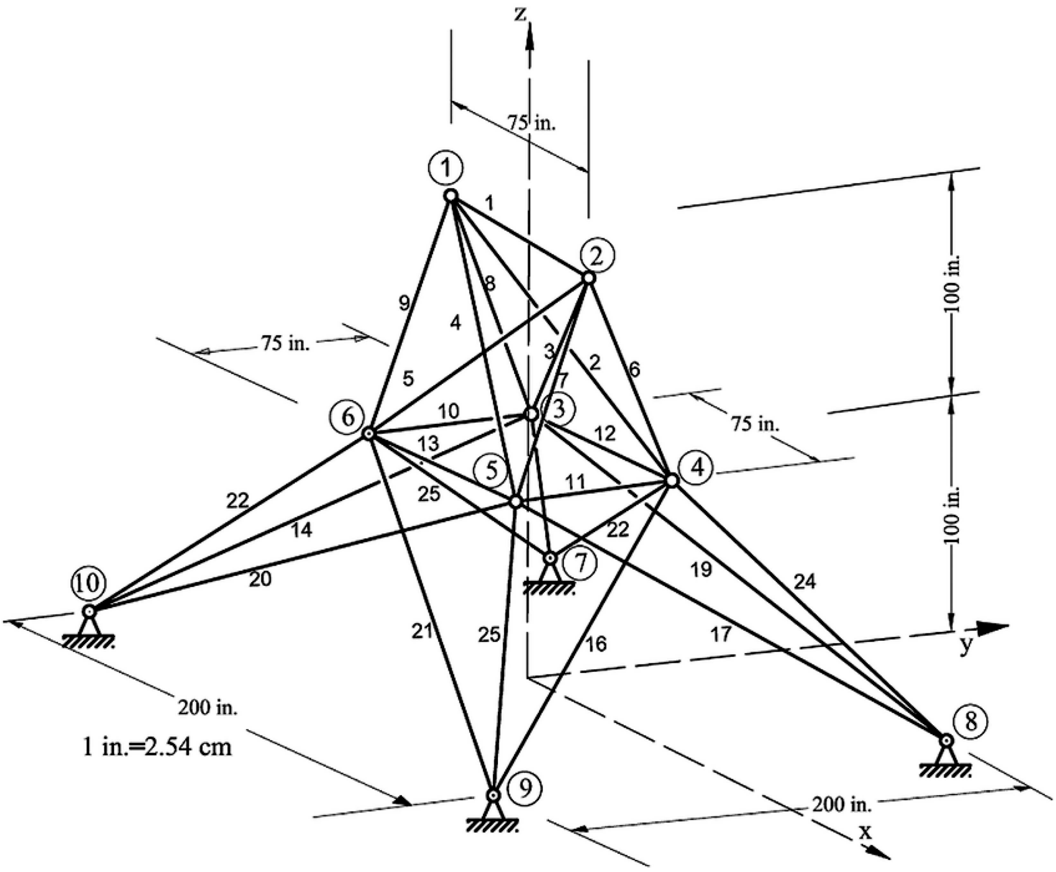


Figure 3. Schematic of the 25-bar spatial truss.

Table 1. Allowable stress values for the spatial 25-bar truss.

Design variables A_i (in^2)	Allowable compressive stress (ksi)	Allowable tension stress (ksi)
A_1	35.092	40.0
$A_2 - A_5$	11.59	40.0
$A_6 - A_9$	17.305	40.0
$A_{10} - A_{11}$	35.092	40.0
$A_{12} - A_{13}$	35.092	40.0
$A_{14} - A_{17}$	6.759	40.0
$A_{18} - A_{21}$	6.959	40.0
$A_{22} - A_{25}$	11.082	40.0

of BBO, ICA, and other algorithms reported in other studies. As observed, MBICA yielded the best weight of 545.08 lb, meaning that the new method was more efficient than other algorithms. In addition, MBICA has a better average weight and standard deviation than the previously proposed algorithms. Figure 4 shows the average convergence histories for MBICA, standard BBO, and ICO through 30 independent runs. It should be noted that MBICA outperforms the BBO and ICA in providing better solutions with the same number of structural analyses.

3.2. 72-bar space truss

Minimizing the weight of the 72-bar spatial truss is the second structural optimization problem investigated in this study. The density and elasticity modulus

Table 2. Loading conditions for the spatial 25-bar truss.

	Condition 1			Condition 2		
Node	F_x	F_y	F_z	F_x	F_y	F_z
1	0.0	20.0	–5.0	1.0	10.0	–5.0
2	0.0	–20.0	–5.0	0.0	10.0	–5.0
3	0.0	0.0	0.0	0.5	0.0	0.0
6	0.0	0.0	0.0	0.5	0.0	0.0

Note: Loads are in kips.

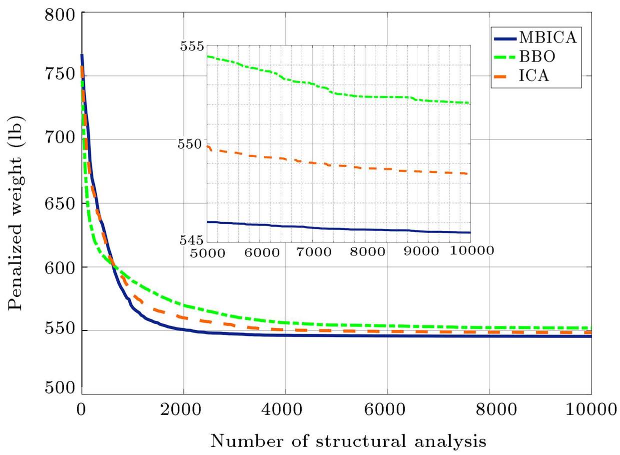


Figure 4. Convergence history curves for the 25-bar truss.

of the used material were obtained as 0.1 lb/in^3 and 10,000 ksi, respectively. Table 4 lists the values and direction of the two loading conditions applied to the truss.

The cross-sections of the members of the design

Table 4. Loading conditions for the spatial 72-bar truss.

	Condition 1			Condition 2		
Node	F_x	F_y	F_z	F_x	F_y	F_z
17	5.0	5.0	–5.0	0.0	0.0	–5.0
18	0.0	0.0	0.0	0.0	0.0	–5.0
19	0.0	0.0	0.0	0.0	0.0	–5.0
20	0.0	0.0	0.0	0.0	0.0	–5.0

Note: Loads are in kips.

variables are divided into 16 groups (Figure 5). The value of 25 ksi is indicative of the allowable stress for all members in the tension and compression states. In both x and y directions, the maximum displacement of free nodes must be less than 0.25 in. The minimum and maximum allowable cross-sections of each member for this structure are limited to 0.10 in^2 and 4.00 in^2, respectively.

The proposed combined method provides minimum weight, average weight, and standard deviation results that are summarized in Table 5. In addition, the MBICA approach requires a smaller number of analyses for convergence than other methods. The average convergence curves are presented in Figure 6 where despite the high convergence rate of the BBO during the initial iterations, the convergence rate of MBICA increases as the number of subpopulations decreases, hence better solutions.

3.3. 120-bar dome truss

Figure 7 explains the support conditions and geometry of the 120-bar dome truss. Accordingly, 120

Table 3. Comparison of optimization results obtained by MBICA and other metaheuristic methods in the 25-bar truss problem.

Study	Kaveh and Talatahari [39]	Talatahari et al. [40]	Talatahari et al. [40]	Talatahari et al. [41]	Jalili and Hosseinzadeh [42]	Present study		
Optimization algorithm	HBB-BC	OICA	CICA-1	MSPSO	BBO-DE	BBO	ICA	MBICA
Element group	Optimal cross-sectional areas (in^2)							
1	0.010	0.010	0.010	0.010	0.010	0.010	0.010	0.010
2	1.993	2.204	1.941	1.985	2.025	2.047	1.934	2.018
3	3.056	2.909	3.035	2.996	3.056	3.118	3.170	3.017
4	0.010	0.010	0.010	0.010	0.010	0.010	0.010	0.010
5	0.010	0.010	0.010	0.010	0.010	0.010	0.010	0.010
6	0.665	0.742	0.661	0.685	0.684	0.641	0.660	0.679
7	1.642	1.538	1.700	1.678	1.613	1.571	1.644	1.638
8	2.679	2.641	2.676	2.660	2.660	2.714	2.655	2.671
Best weight (lb)	545.160	545.931	545.380	545.160	545.09	545.584	545.450	545.081
Average weight (lb)	545.660	549.921	548.532	546.030	545.34	552.052	548.482	545.180
Standard deviation	0.367	3.746	2.701	0.800	0.36	6.189	3.206	0.203
NSAs	12500	5000	5000	10800	13,600	10000	10000	10000

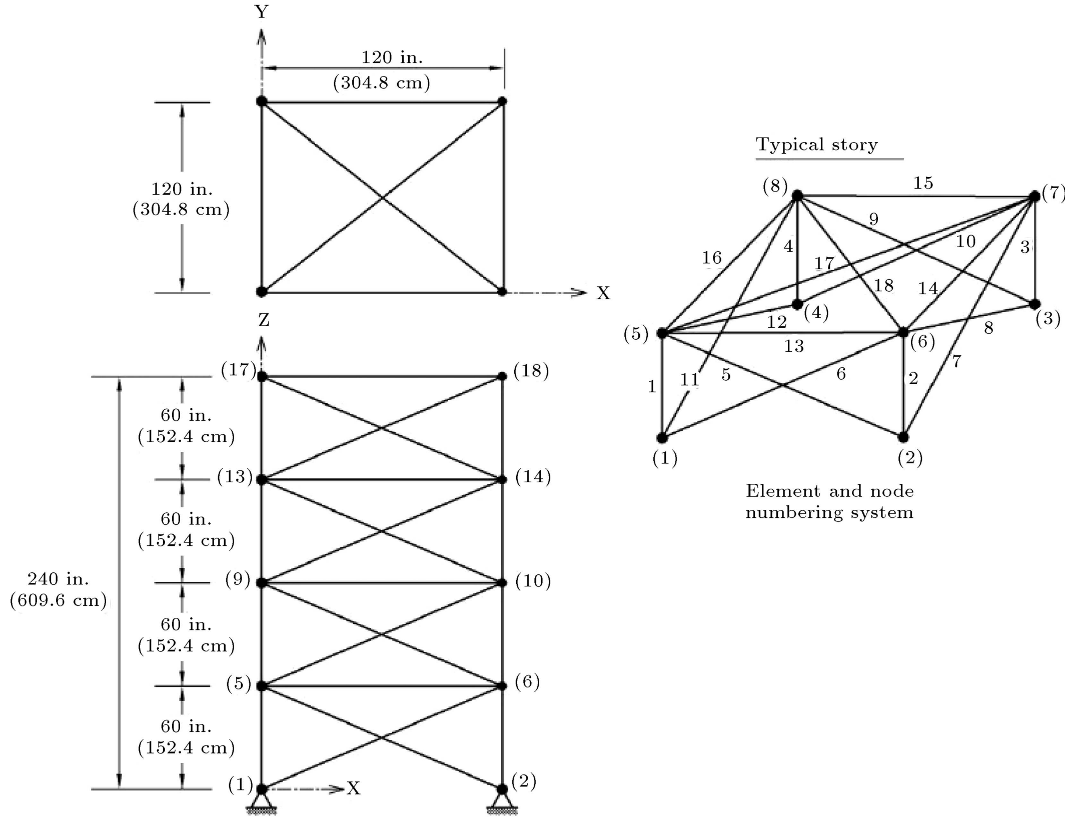


Figure 5. Schematic of the 72-bar spatial truss.

Table 5. Optimal design comparison for the 72-bar truss.

Study	Kaveh and Talatahari [39]	Degertekin [43]	Jalili and Hosseinzadeh [42]	Jafari et al. [44]	Jafari et al. [21]	Present study		
Optimization algorithm	HBB-BC	SAHS	BBO-DE	EHOC	PSOC	BBO	ICA	MBICA
Element group	Optimal cross-sectional areas (in^2)							
1	1.904	1.860	1.901	1.891	1.873	1.928	1.833	1.887
2	0.516	0.521	0.511	0.505	0.513	0.518	0.526	0.512
3	0.100	0.100	0.100	0.100	0.100	0.100	0.100	0.100
4	0.100	0.100	0.100	0.100	0.100	0.100	0.100	0.100
5	1.258	1.271	1.276	1.278	1.276	1.287	1.253	1.274
6	0.503	0.509	0.513	0.508	0.513	0.533	0.511	0.509
7	0.100	0.100	0.100	0.100	0.100	0.100	0.100	0.100
8	0.100	0.100	0.100	0.100	0.100	0.100	0.100	0.100
9	0.518	0.485	0.518	0.530	0.519	0.557	0.502	0.523
10	0.521	0.501	0.517	0.534	0.513	0.484	0.497	0.517
11	0.100	0.100	0.100	0.100	0.100	0.100	0.100	0.100
12	0.101	0.100	0.100	0.100	0.100	0.100	0.100	0.100
13	0.157	0.168	0.157	0.157	0.156	0.157	0.157	0.156
14	0.542	0.584	0.543	0.543	0.560	0.540	0.584	0.548
15	0.413	0.433	0.405	0.407	0.389	0.383	0.421	0.407
16	0.576	0.520	0.571	0.550	0.568	0.597	0.535	0.573
Best weight (lb)	379.66	380.620	379.63	379.72	379.68	380.220	380.462	379.629
Average weight (lb)	381.85	382.420	379.89	380.39	380.48	382.180	384.855	379.797
Standard deviation	1.201	1.380	0.18	0.54	0.58	2.279	4.358	0.084
NSAs	13200	13742	11600	8400	8050	8000	8000	8000

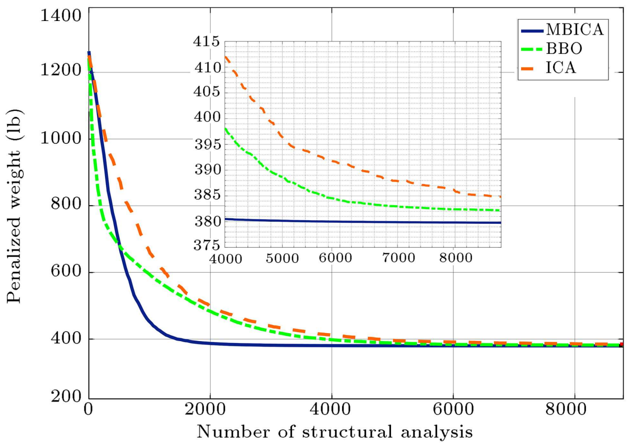
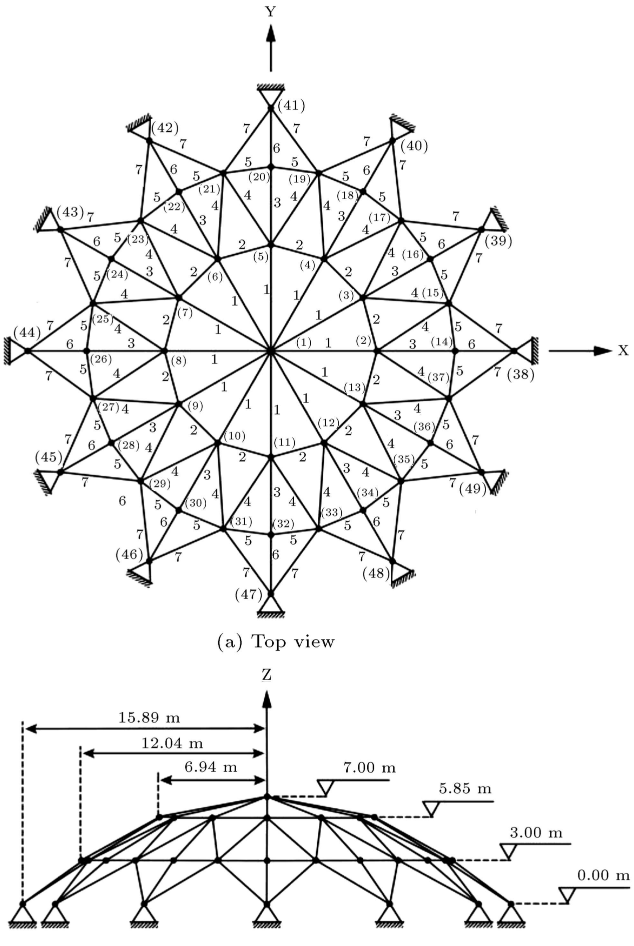


Figure 6. Convergence history curves for the 72-bar spatial truss.



(a) Top view

(b) Side view

Figure 7. Schematic of the 120-bar dome truss.

elements are classified into seven groups considering their structural symmetry. The loading conditions of this structure are considered as 13.49 kips at Node 1, −6.744 kips at Nodes 2 − 14, and −2.248 kips in the remaining nodes. In the case of the design variables, the upper and lower ranges are 0.775 and 20.0 in^2, respectively. According to the AISC ASD (1989) [25],

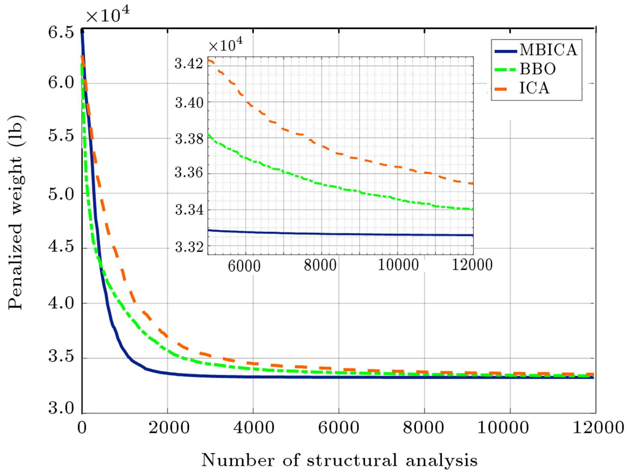


Figure 8. Convergence history curves for the 120-bar dome truss.

the allowable tensile and compressive stresses are as follows:

$$\begin{cases} \sigma_i^+ = 0.6F_y & \text{for} \quad \sigma_i \geq 0 \\ \sigma_i^- & \text{for} \quad \sigma_i < 0 \end{cases}$$

$$\sigma_i^- = \begin{cases} \left[\left(1 - \frac{\lambda_i^2}{2C_c^2}\right)\right] / \left(\frac{5}{3} + \frac{3\lambda_i}{8C_c} - \frac{\lambda_i^3}{8C_c^3}\right) & \text{for} \quad \lambda_i \leq C_c \\ \frac{12\pi^2 E}{23\lambda_i^2} & \text{for} \quad \lambda_i > C_c \end{cases} \tag{19}$$

where E represents the elasticity modulus, F_y the steel yield stress, C_c the slenderness ratio dividing the elastic and inelastic buckling zones ($C_c = \sqrt{2\pi^2 E/F_y}$), and λ_i the slenderness ratio ($\lambda_i = kL_i/r_i$). In addition, the effective length factor is denoted by k (for all truss members, k is set to 1), the member length by L_i, and the minimum gyration radius by r_i. The elasticity modulus is obtained as 30450 ksi, and material density as 0.288 lb/in^3, and steel yield stress as 58.0 ksi. According to the cross-sectional areas, the gyration radius (r_i) can be calculated as $r_i = aA_i^b$ [33]. Here, a and b are fixed, depending on the sections used for the members, such as pipes, angles, and tees. In this example, pipe sections ($a = 0.4993$ and $b = 0.6777$) are used for the bars.

Table 6 reports the results of MBICA and other investigated optimization methods. The proposed hybrid algorithm exhibited the best performance regarding average, best weight, and standard deviation after 8000 analyses, indicating the significant convergence speed of the proposed algorithm in comparison with other metaheuristics. Figure 8 presents the convergence curves for MBICA, BBO, and ICA. As observed, MBICA is capable of generating relatively better solutions with an acceptable convergence rate.

Table 6. Optimal design comparison for the 120-bar dome truss.

Study	Kaveh and Talatahari [45]	Kaveh and Talatahari [39]	Talatahari et al. [41]	Kaveh et al. [46]	Jafari et al. [21]	Present study		
Optimization algorithm	PSACO	HBB-BC	MSPSO	ICHHO	PSOC	BBO	ICA	MBICA
Element group	Optimal cross-sectional areas (in^2)							
1	3.026	3.037	3.024	3.024	3.024	3.027	3.028	3.024
2	15.222	14.431	14.780	14.876	14.776	15.160	15.113	14.732
3	4.904	5.130	5.057	5.009	5.049	5.058	5.058	5.032
4	3.123	3.134	3.136	3.132	3.128	3.112	3.129	3.136
5	8.341	8.591	8.483	8.487	8.488	8.259	8.280	8.506
6	3.418	3.377	3.310	3.299	3.343	3.406	3.385	3.347
7	2.498	2.500	2.498	2.497	2.496	2.513	2.496	2.496
Best weight (lb)	33263.900	33287.900	33251.220	33249.46	33249.43	33281.739	33269.332	33248.990
Average weight (lb)	N/A	N/A	33257.290	33259.49	33261.56	33404.247	33544.154	33258.789
Standard deviation	N/A	N/A	4.290	8.29	11.32	138.271	159.844	4.493
NSAs	32600	10000	15000	15000	6800	12000	12000	12000

3.4. 272-bar transmission tower

The fourth design problem presents a 272-bar transmission tower whose geometric characteristics are depicted in Figure 9. Kaveh and Massoudi [34] introduced this problem for multi-objective optimization and recently, it has been characterized for single-objective optimization to minimize the volume of the structure [35,36]. Based on the structural symmetry, the 272-bar transmission tower members were divided into 28 element groups. The structure is imposed to 12 load cases (Table 7). The nodal displacement for Nodes 1, 2, 11, 20, and 29 is restricted to 100 mm in both x and y directions and to 20 mm in the z direction. The elasticity modulus and permitted stress values for all elements are equal to 2×10^8 kN/m^2 and 275,000 kN/m^2, respectively. Moreover, the lower and upper boundaries of the design variable are 1000 mm^2 and 16000 mm^2.

Table 8 presents the results from the MBICA, BBO, ICA, and two other methods analyzed in the previous studies. According to the findings, the hybrid method yielded a lower total volume than other methods. As observed in Table 8, the average volume and standard deviation are superior to those reported by other different algorithms. The convergence histories of the mean penalized volume of 30 runs for MBICA, BBO, and ICA are illustrated in Figure 10. With the same number of structural analyses, MBICA clearly outperformed both BBO and ICA in terms of the yielded results.

3.5. 582-bar tower truss

Figure 11 presents the fifth design example which is a 582-member truss tower with the height of 80 m. The case study elaborates the application of the discrete design variables for truss structures selected from a study conducted by Hasançebi et al. [37]. Based on the structural symmetry, the members were divided into 32 size variables. The variables in this case are the cross-sectional areas that help minimize the total volume of the tower. A single-load case acts on all the free nodes of the tower including 5 kN for both x and y directions and −30 kN the z direction. The members were selected from a list of 140 W-shaped profiles to determine their cross-sectional areas. For each element, the allowable compressive and tensile stress values were calculated based on the AISC ASD (1989) [38] code. The displacements of all nodes were restricted to 3.15 in in each direction. In addition, the maximum slenderness ratio for the tension and compression members is limited to 300 and 200, respectively.

Table 9 confirms that the MBICA outperforms all

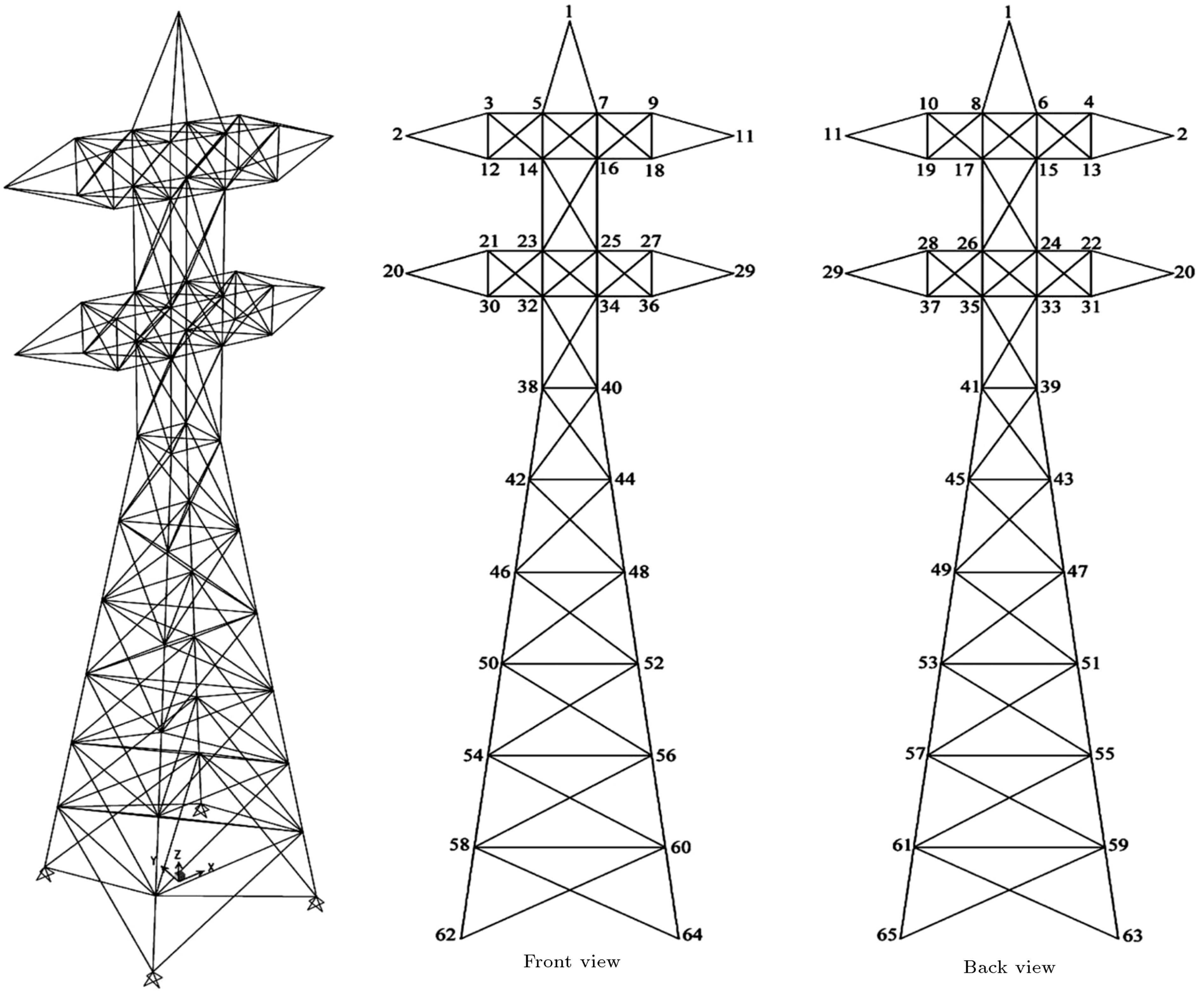


Figure 9. Schematic of the 272-bar transmission tower.

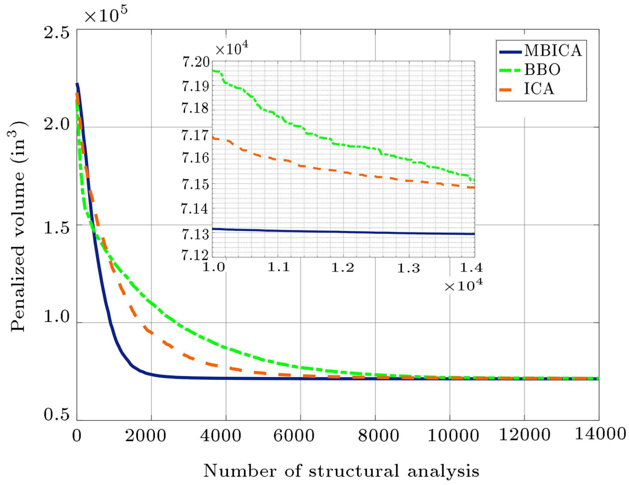


Figure 10. Convergence history curves for the 272-bar transmission tower.

other methods in terms of the volume, average volume, and standard deviation. Figure 12 shows the average convergence curves as well as the high convergence rate of the BBO. However, it is trapped in local minima with unfavorable results, while MBICA is capable of finding better solutions. It can be concluded that upon increasing the structure size, the ability of the proposed algorithm to obtain better solutions becomes more evident than other algorithms.

4. Discussion on the MBICA efficiency

The results from different structural optimization examples indicated that MBICA significantly outperformed the standard BBO, ICA, and some other metaheuristics. The metaheuristic methods employ two critical factors to search for optimum solutions, that is, searching the feasible solution space and using the information gathered throughout the optimization process. In initial iterations, random search serves a more substantial role than the collective data. Despite this fact, the random search power is slowly diminished, and the collective information power is progressively enhanced as the iterations increase. The proposed algorithm provides an appropriate balance between exploitation and exploration using ICA as

Table 7. Loading conditions for the spatial 272-bar transmission tower.

Case	Force direction	Nodes 1	2	11	20	29	Other free nodes
1	F_x (kN)	20	20	20	20	20	5
	F_y (kN)	20	20	20	20	20	5
	F_z (kN)	–40	–40	–40	–40	–40	0
2	F_x (kN)	0	20	20	20	20	5
	F_y (kN)	0	20	20	20	20	5
	F_z (kN)	0	–40	–40	–40	–40	0
3	F_x (kN)	20	0	20	20	20	5
	F_y (kN)	20	0	20	20	20	5
	F_z (kN)	–40	0	–40	–40	–40	0
4	F_x (kN)	20	20	20	0	20	5
	F_y (kN)	20	20	20	0	20	5
	F_z (kN)	–40	–40	–40	0	–40	0
5	F_x (kN)	20	0	0	0	0	5
	F_y (kN)	20	0	0	0	0	5
	F_z (kN)	–40	0	0	0	0	0
6	F_x (kN)	0	20	0	0	0	5
	F_y (kN)	0	20	0	0	0	5
	F_z (kN)	0	–40	0	0	0	0
7	F_x (kN)	0	0	0	20	0	5
	F_y (kN)	0	0	0	20	0	5
	F_z (kN)	0	0	0	–40	0	0
8	F_x (kN)	0	0	20	20	20	5
	F_y (kN)	0	0	20	20	20	5
	F_z (kN)	0	0	–40	–40	–40	0
9	F_x (kN)	0	20	20	0	20	5
	F_y (kN)	0	20	20	0	20	5
	F_z (kN)	0	–40	–40	0	–40	0
10	F_x (kN)	0	0	20	0	20	5
	F_y (kN)	0	0	20	0	20	5
	F_z (kN)	0	0	–40	0	–40	0
11	F_x (kN)	0	0	0	20	20	5
	F_y (kN)	0	0	0	20	20	5
	F_z (kN)	0	0	0	–40	–40	0
12	F_x (kN)	0	0	20	20	0	5
	F_y (kN)	0	0	20	20	0	5
	F_z (kN)	0	0	–40	–40	0	0

Table 8. Optimal design comparison for the 272-bar transmission tower.

Study	Kaveh and Zaerreza [35]	Kaveh et al. [36]	Present study		
Optimization algorithm	**SSOA**	**PGO**	**BBO**	**ICA**	**MBICA**
Element group	**Optimal cross-sectional areas (in^2)**				
1	1.551	1.551	1.550	1.550	1.550
2	1.922	1.886	1.846	1.943	1.931
3	3.862	3.790	3.908	3.622	3.792
4	1.578	1.550	1.560	1.555	1.570
5	14.909	14.849	14.360	15.321	14.858
6	1.550	1.550	1.550	1.550	1.550
7	18.699	19.002	17.847	19.443	18.866
8	1.553	1.552	1.550	1.550	1.552
9	1.550	1.550	1.550	1.550	1.551
10	1.551	1.550	1.550	1.550	1.550
11	15.836	16.455	16.268	15.669	15.982
12	1.550	1.550	1.550	1.550	1.550
13	1.550	1.550	1.550	1.550	1.550
14	1.550	1.550	1.550	1.550	1.550
15	14.447	14.134	14.512	13.832	14.375
16	1.550	1.550	1.550	1.550	1.550
17	1.550	1.550	1.550	1.550	1.550
18	1.554	1.550	1.550	1.550	1.553
19	13.004	13.239	13.992	13.121	13.253
20	1.551	1.550	1.550	1.550	1.550
21	1.550	1.550	1.550	1.550	1.552
22	1.555	1.550	1.550	1.550	1.550
23	12.373	12.382	12.351	13.341	12.478
24	1.551	1.550	1.550	1.550	1.550
25	1.551	1.550	1.550	1.550	1.550
26	1.550	1.550	1.550	1.550	1.550
27	11.632	11.868	12.397	11.937	11.696
28	1.550	1.550	1.550	1.550	1.550
Best volume (in^3)	71287.976	71282.663	71338.801	71341.350	71278.648
Average volume (in^3)	71316.541	71324.151	71513.099	71484.466	71294.309
Standard deviation	18.963	47.082	165.354	107.411	8.466
NSAs	14,020	23,920	14000	14000	14000

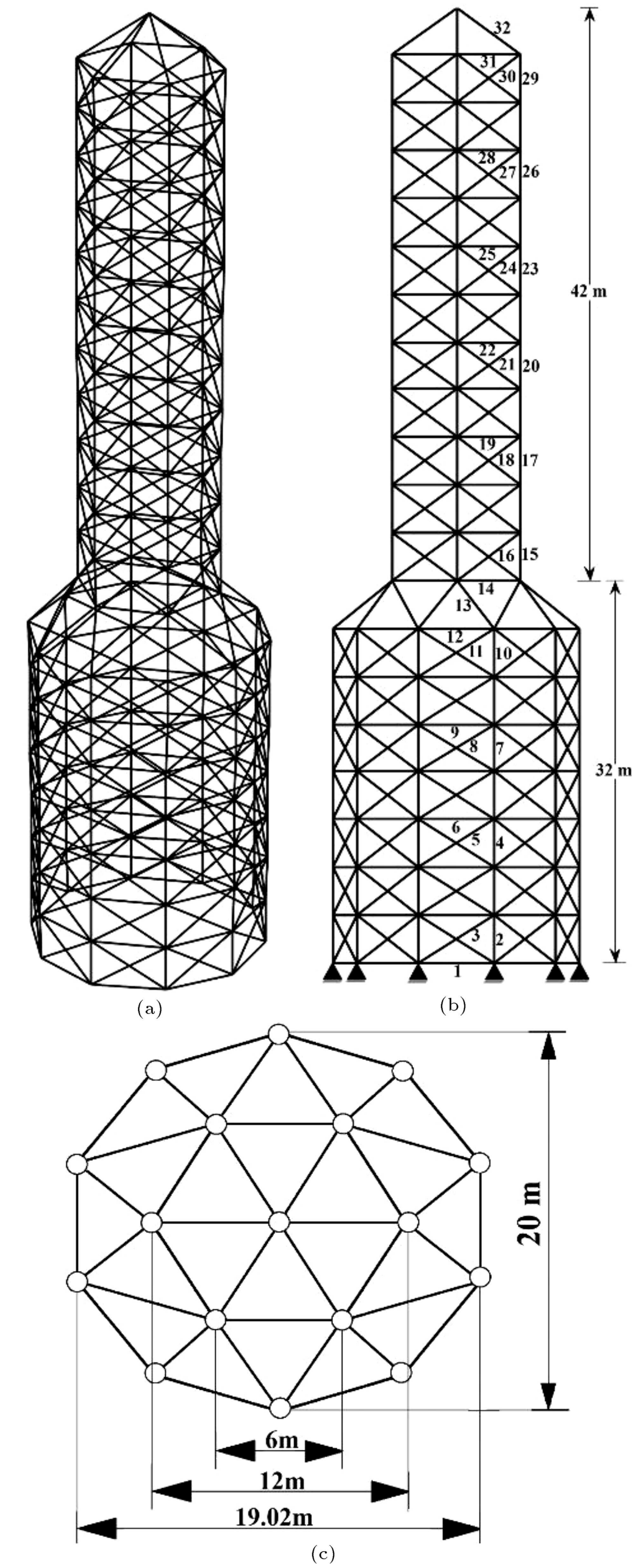


Figure 11. Schematic of the 582-bar tower truss.

a global optimizer for global exploration and BBO migration operator for local exploitation. In the MBICA, algorithm diversification in the early stages occurs by dividing the population of solutions into

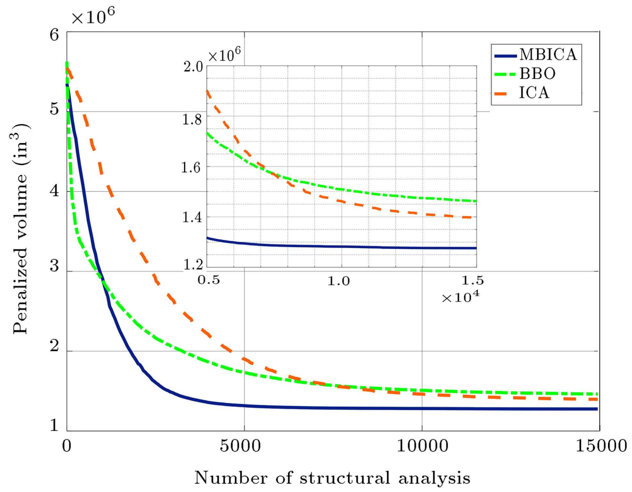


Figure 12. Convergence history curves for the 582-bar tower.

smaller subpopulations. In addition, there are often local optimum solutions near the desirable solutions in complex optimization problems such as structural optimization. Hence, further investigation into the local optimum increases the probability of finding a satisfying optimum solution.

The proposed method conducts additional searches around the local optimum solutions to achieve the desired solution in fewer analyses using the migration operator. Moreover, as the search procedure continues, the number of subpopulations decreases and the intensification ability of the algorithm is enhanced accordingly.

5. Conclusion

The present research proposed a new hybrid algorithm with the basis of the Biogeography-Based Optimization (BBO) and Imperialist Competitive Algorithm (ICA). The newly developed optimization algorithm, called the Migration-Based Imperialist Competitive Algorithm (MBICA), can determine global optima through relatively fewer structural analyses. This method enjoys the combined advantages of both BBO and ICA techniques while avoiding their weaknesses. Five design examples were examined to assess the performance of the proposed algorithm. The hybrid method yielded better results with lower standard deviations than the standard versions of the ICA, BBO, and some other metaheuristics described in the literature. The results indicated that despite the higher convergence rate of the BBO in the initial stages, it could be potentially trapped in local minima. As the number of subpopulations decreased by progressing the optimization process, MBICA could reach a satisfactory convergence rate and yield more favorable results. With the same number of structural analyses, the hybrid method had a higher convergence rate than

Table 9. Optimal design comparison for the 582-bar tower.

Study	Kaveh and Talatahari [47]	Kaveh and Talatahari [48]	Mortazavi and Toğan [49]	Kaveh and Ilchi [50]	Jalili and Hosseinzadeh [42]	Present study		
Optimization algorithm	DHPSACO	HBB-BC	iPSO	UECBO	BBO-DE	BBO	ICA	MBICA
Element group	Optimal W-shaped sections							
1	W8X24	W8X24	W8X21	W8X21	W12X22	W8X21	W8X21	W8X21
2	W12X72	W24X68	W8X21	W14X90	W24X76	W12X79	W10X88	W14X90
3	W8X28	W8X28	W8X21	W8X24	W8X28	W8X28	W8X24	W8X24
4	W12X58	W18X60	W21X73	W14X61	W12X58	W14X48	W12X65	W10X60
5	W8X24	W8X24	W12X53	W8X24	W10X30	W8X24	W8X24	W8X24
6	W8X24	W8X24	W8X21	W8X21	W14X22	W8X21	W8X21	W8X21
7	W10X49	W21X48	W8X21	W14X48	W14X48	W12X53	W24X131	W12X45
8	W8X24	W8X24	W8X21	W8X24	W8X24	W8X24	W10X22	W8X24
9	W8X24	W10X26	W8X21	W8X21	W8X21	W8X21	W8X21	W8X21
10	W12X40	W14X38	W8X21	W14X43	W18X50	W14X48	W12X58	W14X48
11	W12X30	W12X30	W18X76	W8X24	W10X22	W8X21	W8X21	W8X21
12	W12X72	W12X72	W24X62	W16X67	W18X55	W21X73	W14X82	W12X58
13	W18X76	W21X73	W10X49	W12X72	W21X73	W12X87	W12X65	W14X74
14	W10X49	W14X53	W10X49	W12X50	W16X67	W12X53	W10X77	W12X53
15	W14X82	W18X86	W12X79	W18X76	W14X74	W16X89	W8X35	W18X76
16	W8X31	W8X31	W21X62	W8X31	W8X31	W8X31	W8X35	W8X31
17	W14X61	W18X60	W14X43	W10X60	W14X61	W16X36	W16X50	W10X68
18	W8X24	W8X24	W16X26	W8X24	W8X24	W6X25	W12X26	W10X22
19	W8X21	W16X36	W8X21	W8X21	W14X22	W8X31	W8X21	W8X21
20	W12X40	W10X39	W8X21	W12X45	W16X40	W12X45	W16X36	W10X45
21	W8X24	W8X24	W8X21	W8X24	W8X21	W8X24	W8X21	W8X21
22	W14X22	W8X24	W8X24	W8X21	W8X21	W33X118	W8X21	W8X21
23	W8X31	W8X31	W8X24	W8X21	W8X31	W12X30	W8X24	W6X25
24	W8X28	W8X28	W8X21	W8X24	W8X21	W8X21	W8X21	W8X21
25	W8X21	W8X21	W16X67	W8X21	W10X22	W8X21	W8X21	W8X21
26	W8X21	W8X24	W8X31	W8X21	W12X22	W10X22	W8X21	W8X21
27	W8X24	W8X28	W8X24	W8X24	W10X22	W14X22	W8X21	W8X21
28	W8X28	W14X22	W8X21	W8X21	W12X22	W8X21	W8X21	W8X21
29	W16X36	W8X24	W8X21	W8X21	W10X22	W14X34	W8X21	W8X21
30	W8X24	W8X24	W8X21	W8X24	W10X22	W14X22	W8X21	W8X21
31	W8X21	W14X22	W8X21	W8X21	W8X21	W8X21	W8X21	W8X21
32	W8X24	W8X24	W8X28	W8X24	W10X22	W8X21	W8X21	W8X21
Best volume (in^3)	1346227	1365143	1278228	1294929	1318112	1395428	1362127	1267616
Average volume (in^3)	N/A	N/A	1301618	1311709	1351675	1462741	1396833	1275971
Standard deviation	N/A	N/A	58296	N/A	15255	27702	21748	2464
NSAs	8500	12500	2360	4090	17540	15000	15000	15000

that of the ICA and outperformed it in obtaining optimized solutions. Based on the obtained results, it can be concluded that the proposed method was effective, efficient, and robust. The future MBICA application as a metaheuristic algorithm can be even more encouraging. In addition to other structures such as frame structures, it can also be applied to many other engineering problems.

References

1. Mirjalili, S., Mirjalili, S.M., and Lewis, A. "Grey wolf optimizer", *Advances in Engineering Software*, **69**, pp. 46–61 (2014). DOI: 10.1016/j.advengsoft.2013.12.007
2. Kaveh, A., *Advances in Metaheuristic Algorithms for Optimal Design of Structures*, Springer Nature (2021). DOI: 10.1007/978-3-030-59392-6
3. Holland, J.H. "Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence", MIT Press (1992).
4. Storn, R. and Price, H. "Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces", *Journal of Global Optimization*, **11**(4), pp. 341–59 (1997). DOI: 10.1023/A:1008202821328
5. Kennedy, J. and Eberhart, R. "Particle swarm optimization", *Proceedings of ICNN'95-International Conference on Neural Networks: IEEE*. pp. 1942–8 (1995).
6. Dorigo, M., Maniezzo, V., and Colorni, A. "Ant system: optimization by a colony of cooperating agents", *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, **26**(1), pp. 29–41 (1996). DOI: 10.1109/3477.484436
7. Erol, O.K. and Eksin, I. "A new optimization method: big bang-big crunch", *Advances in Engineering Software*, **37**(2), pp. 106–11 (2006). DOI: 10.1016/j.advengsoft.2005.04.005
8. Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P. "Optimization by simulated annealing", *Science*, **220**(4598), pp. 671–80 (1983). DOI: 10.1126/science.220.4598.671
9. Kaveh, A. and Talatahari, S. "A novel heuristic optimization method: charged system search", *Acta Mechanica*, **213**(3), pp. 267–89 (2010). DOI: 10.1007/s00707-009-0270-4
10. Kaveh, A. and Mahdavi, V.R. "Colliding bodies optimization: a novel meta-heuristic method", *Computers & Structures*, **139**, pp. 18–27 (2014). DOI: 10.1016/j.compstruc.2014.04.005
11. Kaveh, A. and Eslamlou, A.D., *Metaheuristic Optimization Algorithms in Civil Engineering: New Applications*, Springer Nature (2020). DOI: 10.1007/978-3-030-45473-9
12. Kaveh, A. and Rajabi, F. "Fuzzy-multi-mode resource-constrained discrete time-cost-resource optimization in project scheduling using ENSCBO", *Periodica Polytechnica Civil Engineering*, **66**(1), pp. 50–62 (2022). DOI: 10.3311/PPci.19145
13. Kaveh, A., Rajabi, F., and Mirvalad, S. "Many-objective optimization for construction project scheduling using non-dominated sorting differential evolution algorithm based on reference points", *Scientia Iranica, A*, **28**(6), pp. 3112–33128 (2021). DOI: 10.24200/sci.2021.58952.5988
14. Kaveh, A., Hamedani, K.B., and Kamalinejad, M. "An enhanced Forensic-Based Investigation algorithm and its application to optimal design of frequency-constrained dome structures", *Computers & Structures*, **256**, p. 106643 (2021). DOI: 10.1016/j.compstruc.2021.106643
15. Azad, S.K., Aminbakhsh, S., and Shaban, S.S. "Multi-stage guided stochastic search for optimization and standardization of free-form steel double-layer grids", Structures: Elsevier. pp. 678–99 (2021).
16. Kaveh, A., Kamalinejad, M., Arzani, H., et al. "New enhanced colliding body optimization algorithm based on a novel strategy for exploration", *Journal of Building Engineering*, **43**, p. 102553 (2021). DOI: 10.1016/j.jobe.2021.102553
17. Azad, S.K. "Monitored convergence curve: a new framework for metaheuristic structural optimization algorithms, structural and multidisciplinary optimization", *Structural and Multidisciplinary Optimization*, **60**(2), pp. 481–99 (2019). DOI: 10.1007/s00158-019-02219-5
18. Kaveh, A., Moghanni, R.M., and Javadi, S. "Chaotic optimization algorithm for performance-based optimization design of composite moment frames", *Engineering with Computers*, pp. 1–13 (2021). DOI: 10.1007/s00366-020-01244-z
19. Azad, S.K. and Hasançebi, O. "Upper bound strategy for metaheuristic based design optimization of steel frames", *Advances in Engineering Software*, **57**, pp. 19–32 (2013). DOI: 10.1016/j.istruc.2021.07.068
20. Ting, T., Yang, X.-S., Cheng, S., et al. "Hybrid metaheuristic algorithms: past, present, and future", *Recent Advances in Swarm Intelligence and Evolutionary Computation*, pp. 71–83 (2015). DOI: 10.1007/978-3-319-13826-8_4
21. Jafari, M., Salajegheh, E., and Salajegheh, J. "Optimal design of truss structures using a hybrid method based on particle swarm optimizer and cultural algorithm", Structures: Elsevier. pp. 391–405 (2021).
22. Le, D.T., Bui, D.-K., Ngo, T.D., et al. "A novel hybrid method combining electromagnetism-like mechanism and firefly algorithms for constrained design optimization of discrete truss structures", *Computers & Structures*, **212**, pp. 20–42 (2019). DOI: 10.1016/j.compstruc.2018.10.017
23. Lieu, Q.X., Do, D.T., and Lee, J. "An adaptive hybrid evolutionary firefly algorithm for shape and size optimization of truss structures with frequency

constraints", *Computers & Structures*, **195**, pp. 99–112 (2018). DOI: 10.1016/j.compstruc.2017.06.016

24. Sharma, S., Saha, A.K., and Lohar, G. "Optimization of weight and cost of cantilever retaining wall by a hybrid metaheuristic algorithm", *Engineering with Computers*, pp. 1–27 (Aug. 2022). DOI: 10.12989/gae.2021.24.3.237

25. Talatahari, S., Gandomi, A.H., Yang, X.-S., et al. "Optimum design of frame structures using the eagle strategy with differential evolution", *Engineering Structures*, **91**, pp. 16–25 (2015). DOI: 10.1016/j.engstruct.2015.02.026

26. Kaveh, A. and Ilchi Ghazaan, M. "A new hybrid meta-heuristic algorithm for optimal design of large-scale dome structures", *Engineering Optimization*, **50**(2), pp. 235–52 (2018). DOI: 10.1080/0305215X.2017.1313250

27. Simon, D. "Biogeography-based optimization", *IEEE Transactions on Evolutionary Computation*, **12**(6), pp. 702–13 (2008). DOI: 10.1109/TEVC.2008.919004

28. Atashpaz-Gargari, E. and Lucas, C. "Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition", *IEEE Congress on Evolutionary Computation: Ieee.*, pp. 4661–7 (2007).

29. Kaveh, A. and Bakhshpoori, T. "Metaheuristics: outlines, MATLAB codes and examples", Springer (2019). DOI: 10.1007/978-3-030-04067-3

30. Talbi, E.-G. "A taxonomy of hybrid metaheuristics", *Journal of Heuristics*, **8**(5), pp. 541–64 (2002). DOI: 10.1023/A:1016540724870

31. Jalili, S., Hosseinzadeh, Y., and Taghizadieh, N. "A biogeography-based optimization for optimum discrete design of skeletal structures", *Engineering Optimization*, **48**(9), pp. 1491–514 (2016). DOI: 10.1080/0305215X.2015.1115028

32. Kaveh, A. and Talatahari, S. "Optimum design of skeletal structures using imperialist competitive algorithm", *Computers & Structures*, **88**(21–22), pp. 1220–9 (2010). DOI: 10.1016/j.compstruc.2010.06.011

33. Saka, M. "Optimum design of pin-jointed steel structures with practical applications", *Journal of Structural Engineering*, **116**(10), pp. 2599–620 (1990). DOI: 10.1061/(ASCE)0733-9445(1990)116:10(2599)

34. Kaveh, A. and Massoudi, M. "Multi-objective optimization of structures using charged system search", *Scientia Iranica*, **21**(6), pp. 1845–60 (2014).

35. Kaveh, A. and Zaerreza, A. "Shuffled shepherd optimization method: a new meta-heuristic algorithm", *Engineering Computations*, **37**(7), pp. 2357–89 (2020). DOI: 10.1108/ec-10-2019-0481

36. Kaveh, A., Akbari, H., and Hosseini, S.M. "Plasma generation optimization: a new physically-based meta-heuristic algorithm for solving constrained optimization problems", *Engineering Computations*, **38**(4), pp. 1554–606 (2020). DOI: 10.1108/ec-05-2020-0235

37. Hasançebi, O., Çarbaş, S., Doğan, E., et al. "Performance evaluation of metaheuristic search techniques in the optimum design of real size pin jointed structures", *Computers & Structures*, **87**(5–6), pp. 284–302 (2009). DOI: 10.1016/j.compstruc.2009.01.002

38. Manual of steel construction. Allowable stress design. Chicago, Illinois, USA. 9th Ed, American Institute of Steel Construction (AISC) (1989).

39. Kaveh, A. and Talatahari, S. "Size optimization of space trusses using big bang-big crunch algorithm", *Computers & Structures*, **87**(17–18), pp. 1129–40 (2009). DOI: 10.1016/j.compstruc.2009.04.011

40. Talatahari, S., Kaveh, A., and Sheikholeslami, R. "Chaotic imperialist competitive algorithm for optimum design of truss structures", *Structural and Multidisciplinary Optimization*, **46**(3), pp. 355–67 (2012). DOI: 10.1007/s00158-011-0754-4

41. Talatahari, S., Kheirollahi, M., Farahmandpour, C., et al. "A multi-stage particle swarm for optimum design of truss structures", *Neural Computing and Applications*, **23**(5), pp. 1297–309 (2013). DOI: 10.1007/s00521-012-1072-5

42. Jalili, S. and Hosseinzadeh, Y. "Design optimization of truss structures with continuous and discrete variables by hybrid of biogeography-based optimization and differential evolution methods", *The Structural Design of Tall and Special Buildings*, **27**(14), p. e1495 (2018). DOI: 10.1002/tal.1495

43. Degertekin, S. "Improved harmony search algorithms for sizing optimization of truss structures", *Computers & Structures*, **92**, pp. 229–41 (2012). DOI: 10.1016/j.compstruc.2011.10.022

44. Jafari, M., Salajegheh, E., and Salajegheh, J. "An efficient hybrid of elephant herding optimization and cultural algorithm for optimal design of trusses", *Engineering with Computers*, **35**(3), pp. 781–801 (2019). DOI: 10.1007/s00366-018-0631-5

45. Kaveh, A. and Talatahar, S. "A hybrid particle swarm and ant colony optimization for design of truss structures", *Asian Journal of Civil Engineering*, **9**(4), pp. 325–344 (2008).

46. Kaveh, A., Rahmani, P., and Eslamlou, A.D. "An efficient hybrid approach based on Harris Hawks optimization and imperialist competitive algorithm for structural optimization", *Engineering with Computers*, pp. 1–29, Pub Date: 2021-02-03. DOI: 10.1007/s00366-020-01258-7

47. Kaveh, A. and Talatahari, S. "A particle swarm ant colony optimization for truss structures with discrete variables", *Journal of Constructional Steel Research*, **65**(8–9), pp. 1558–68 (2009). DOI: 10.1016/j.jcsr.2009.04.021

48. Kaveh, A. and Talatahari, S. "A discrete big bang-big crunch algorithm for optimal design of skeletal structures", *Asian Journal of Civil Engineering*, **11**(1), pp. 103–123 (2010).

49. Mortazavi, A. and Toğan, V. "Simultaneous size, shape, and topology optimization of truss structures using integrated particle swarm optimizer", *Structural and Multidisciplinary Optimization*, **54**(4), pp. 715–36 (2016). DOI: 10.1007/s00158-016-1449-7

50. Kaveh, A. and Ilchi, G.M. "A hybrid ecbo and ubs algorithm for optimal design of skeletal structures", *Asian Journal of Civil Engineering*, **17**(7), pp. 918–36 (2016).DOI: 10.24200/j30.2018.1369

Biographies

Ali Kaveh was born in Tabriz, Iran in 1948. After graduation from the Department of Civil Engineering at the University of Tabriz in 1969, he continued his studies in the field of Structures at Imperial College of Science and Technology at London University and received his MSc, DIC, and PhD degrees in 1970 and 1974, respectively. He then joined Iran University of Science and Technology. Professor Kaveh is the author of 730 papers published in international journals and 170 papers presented at national and international conferences. He has authored 23 books in Persian and 16 books in English published by Wiley, Research Studies Press, American Mechanical Society, and Springer.

Farivar Rajabi is a graduate student in the Construction Engineering and Management program of the School of Civil Engineering, Iran University of Science and Technology, Tehran, Iran. He also received his BSc degree in Civil Engineering from Iran University of Science and Technology, Tehran, Iran. His research interests include artificial intelligence applications in design and construction.