Computers and Structures 256 (2021) 106643

Contents lists available at ScienceDirect

Computers and Structures

journal homepage: www.elsevier.com/locate/compstruc

An enhanced Forensic-Based Investigation algorithm and its application to optimal design of frequency-constrained dome structures

Ali Kaveh *, Kiarash Biabani Hamedani, Mohammad Kamalinejad

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

ARTICLE INFO

Article history:
Received 22 December 2020
Accepted 18 July 2021
Available online 7 August 2021

Keywords:
Forensic-Based Investigation
Structural optimization
Truss structures
Natural frequencies
Frequency constraints
Dome

ABSTRACT

In this paper, an Enhanced Forensic-Based Investigation (EFBI) is proposed for optimal design of frequency-constrained dome-like trusses. The Forensic-Based Investigation (FBI) algorithm is a newly developed population-based metaheuristic inspired by the criminal investigation process. Throughout the FBI algorithm, the investigation and pursuit teams (i.e., the diversification and the intensification components of FBI, respectively) cooperate with each other to solve the target optimization problem. This study aims to modify the original formulation of FBI and propose an enhanced version of FBI, named as EFBI. The proposed EFBI seeks to reinforce communication between the investigation and pursuit teams with the aim of striking a better balance between the intensification and the diversification tasks. The performance of the proposed EFBI is compared with those of the standard FBI and some other optimization methods through three well-studied dome truss optimization problems with frequency constraints. To the best of our knowledge, this is the first time that FBI is applied to structural optimization. Results confirm that EFBI significantly outperforms the standard FBI, and has superior or comparable performance to other considered metaheuristics.

1. Introduction

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

Optimum design of structures considering frequency constraints has attracted the attention of many researchers since the 1980s. Bellagamba and Yang [3] applied a nonlinear programming procedure to the minimum mass design of truss structures under static thermal and mechanical loads considering various constraints, including fundamental natural frequency and local buckling. Grandhi and Venkayya [2] employed an optimality criterion method based on uniform Lagrangian density for the minimum weight design of structures with multiple frequency constraints. Tong and Liu [4] proposed a dynamically constrained optimization procedure for the optimal design of truss structures with discrete design variables under dynamic constraints. Sedaghati et al. [5] used the integrated force method as an analyzer to optimize both truss and beam structures under frequency constraints. Lingyun et al. [6] proposed a Niche Hybrid Genetic Algorithm (NHGA) for solving shape and size optimization of truss structures with multiple frequency constraints. Gomes [7] employed a Particle Swarm Optimization (PSO) algorithm for size and shape optimization of truss structures taking into account frequency constraints. Kaveh and Zolghadr [8] proposed a hybridization of the Charged System Search (CSS) and the Big Bang-Big Crunch (BB-BC) algorithms with trap recognition capability and applied it to the optimization of truss structures with frequency constraints. Khatibnia and Naseralavi [1] introduced the Orthogonal Multi-Gravitational Search Algorithm (OMGSA) to solve size and shape truss optimization problems with frequency constraints. Kaveh and Ilchi Ghazaan [9] performed the optimal design of large-scale dome structures with multiple natural frequency constraints. Ho-Huu et al. [10] proposed an improved Differential Evolution (DE) for size and

* Corresponding author at: School of Civil Engineering, Iran University of Science and Technology, Narmak, Tehran, P.O. Box 16846-13114, Iran.
E-mail address: alikaveh@iust.ac.ir (A. Kaveh).

https://doi.org/10.1016/j.compstruc.2021.106643
0045-7949/

shape optimization of truss structures with frequency constraints. Kaveh and Zolghadr [11] utilized the Cyclical Parthenogenesis Algorithm (CPA) for optimal design of cyclically symmetric trusses with frequency constraints. Lieu et al. [12] proposed a hybridization of the Differential Evolution (DE) algorithm and the Firefly Algorithm (FA) for shape and size optimization of truss structures under multiple frequency constraints.

Frequency-constrained optimization problems are well-known as highly nonlinear, non-convex, and multimodal optimization problems with respect to the design variables [13]. The most common difficulty with the frequency-constrained optimization problems is the switching of vibration modes due to structural size and shape modifications, which may cause convergence difficulties [14]. As a result, classical methods of optimization based on gradients may not be effective to solve this type of optimization problems because these methods require the gradient information of the frequency with respect to the design variables [6]. Thus, metaheuristic optimization algorithms can be regarded as appropriate alternatives.

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

Over the past three decades, numerous metaheuristics have been developed, most of which are nature-inspired. Nature-inspired metaheuristic algorithms can be categorized into four main groups (see Fig. 1): evolution-based, physics-based, swarm-based, and human-based [17]. Evolution-based algorithms imitate the concepts of biological evolution. Some of the most popular evolution-based algorithms are Genetic Algorithm (GA) [18], Evolution Strategy (ES) [19], Genetic Programming (GP) [20], Evolutionary Programming (EP) [21], Memetic Algorithm (MA) [22], and Differential Evolution (DE) [23]. Physics-based algorithms are inspired by the physical laws of nature. Some of the most well-known physics-based algorithms are Gravitational Search Algorithm (GSA) [24], Charged System Search (CSS) [25], Big Bang-Big Crunch (BB-BC) [26], Ray Optimization (RO) [27], and Water Cycle Algorithm (WCA) [28]. Swarm-based algorithms form another group of nature-inspired algorithms that are based on the social behavior of groups of animals. Some of the most popular swarm-based metaheuristic algorithms are Artificial Bee Colony (ABC) [29], Particle Swarm Optimization (PSO) [30], Ant Colony Optimization (ACO) [31], and Cuckoo Search (CS) [32]. The last group of nature-inspired algorithms is human-based metaheuristics that mimic the human behaviors and characteristics. Harmony Search (HS) mimics the improvisation of music players [33]. League Championship Algorithm (LCA) simulates the competition of sport teams in a sport league [34]. Teaching-Learning-Based Optimization (TLBO) algorithm is developed based on the philosophy of the teaching-learning process [35]. Imperialist Competitive Algorithm (ICA) is established inspired by the policy of extending a country's beyond its own boundaries [36]. Dynastic Optimization Algorithm (DOA) is developed by using social behavior in human dynasties [37]. Interior Search Algorithm (ISA) is established inspired by interior design and decoration [38].

Recently, the Forensic-Based Investigation (FBI) algorithm is developed based on the suspect investigation-location-pursuit process [39]. There are two teams in the criminal investigation process, one is an investigation team, and the other is a pursuit team. Accordingly, FBI has two phases, i.e., investigation and pursuit phases, each of which has its own distinct population of solutions. In other words, each team continues independently the search process. The teams interact with each other only through the best solution found so far. Poor communication between the investigation and pursuit teams causes the convergence rate of FBI to be decreased.

In this paper, an Enhanced Forensic-Based Investigation (EFBI) is proposed for frequency-constrained truss optimization problems. The Enhanced Forensic-Based Investigation algorithm is designed in order to eliminate the drawback of FBI as mentioned above, and improve the balance between the intensification and the diversification tasks (more effective interaction between the investigation and pursuit teams). To achieve this purpose, the investigation and pursuit teams are assumed to work in close cooperation with each other, so that each team starts its own search process with the output of the previous phase. In order to demonstrate the efficiency and capability of EFBI, three size and shape dome-like truss optimization problems with multiple frequency constraints are considered from literature. The optimization results are compared with those in the literature. The results reveal that the proposed EFBI significantly performs better than the standard FBI in terms of accuracy, reliability, and convergence rate.

The rest of this paper is organized as follows: The standard FBI is reviewed in Section 2. In Section 3, after pointing out the drawbacks of the standard FBI, an enhanced version of the FBI algorithm is proposed. In Section 4, truss optimization problem with natural frequency constraints is stated. Three numerical examples are studied in Section 5. Finally, some concluding remarks are provided in Section 6.

2. Forensic-Based investigation (FBI)[1]

The Forensic-Based Investigation (FBI) is a parameter-free human-based metaheuristic optimization algorithm developed by Chou and Nguyen in 2020 [39]. The algorithm is inspired by the forensic investigation process in criminal cases. The following subsection describes the forensic investigation process.

2.1. Forensic investigation process

The process of large-scale criminal investigations can be seen as a process composed of five steps: start of the investigation, interpretation of the findings, directions of inquiry, actions, and prosecution of a suspect [40]. The middle three steps of the criminal investigation process can be seen as a cycle. Fig. 2 shows steps of the criminal investigation process. This cyclic process starts as soon as a crime is reported and ends when the truth is discovered. Immediately after a crime is reported, a criminal investigation team is formed to start the investigation process. The team investigates the crime scene and the victim with a number of standard procedures. In the first step of the investigation process, initial information about the crime is provided, which is a starting point for the investigators. In the second step, members of the investigation team interpret the findings and share their points of view with each other. Next, based on the interpretation of the findings, the team members suggest several directions of inquiry. The team reviews the findings and sees whether they match with existing directions of inquiry. As a result, new directions of inquiry may be suggested, and existing ones may be changed, ignored, or confirmed. In the next step of a criminal investigation, the team decides about further actions that need to be taken. The investigation

[1] The source code of the original algorithm is available at https://www.researchgate.net/profile/Jui-Sheng_Chou

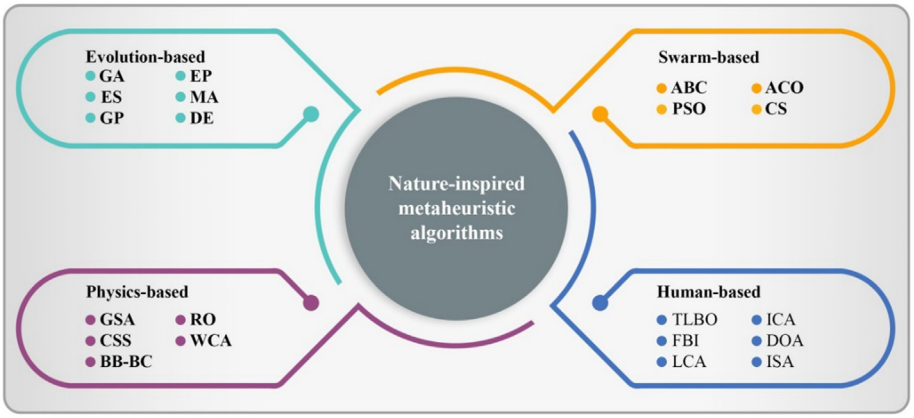

Fig. 1. Classification of nature-inspired metaheuristic algorithms.

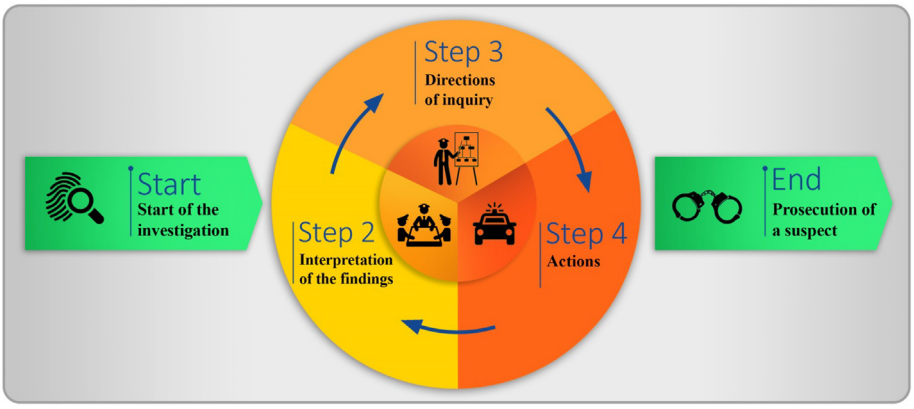

Fig. 2. Steps of the criminal investigation process.

team usually investigates the most promising direction of research first. The actions taken result in new clues and information. As the new findings are collected, the team members interpret them, which may result in new several directions of inquiry and actions. This process continues until the truth is discovered (i.e., when a serious suspect is identified and prosecuted).

2.2. Mathematical model

The standard FBI is designed in two main search phases: the investigation phase and the pursuit phase. The investigation phase deals with the diversification of the search, while the pursuit phase ensures the intensification task. Search space of the algorithm is defined as anywhere the suspect might be hiding, and the probability of locating the suspect serves as a metaphor for the objective function. In the following steps, the optimization problem is defined in the maximization form of the objective function. Thus, the more the value of the objective function means the better chance of locating the suspect. In order to start the criminal investigation process, a number of initial random suspected locations are considered as follows:

$$X_{A_{ij}} = LB_j + rand_{ij}(UB_j - LB_j);\ \ i = 1,2,\cdots,NP \ \text{ and } \ j = 1,2,\cdots,D. \tag{1}$$

where NP represents the number of suspected locations (population size); D is the number of design variables of the problem; LB and UB denote the lower and upper bound vectors of design variables; $rand$ is a random number between 0 and 1; and X_{A_i} determines the i-th possible location of the suspect, which hereafter is called i-th suspected location.

2.2.1. Investigation phase (A)

Investigation phase includes the steps of: (1) interpretation of the findings; and (2) directions of inquiry. Immediately after a crime is detected, a team of investigators is formed to realize what might have happened. The criminal investigation team starts the investigation by use of the information relating to the crime scene. In the next two subsections, the investigation phase is modelled.

2.2.1.1. Interpretation of the findings (A_1). Members of the investigation team interpret their findings and discuss different points of view about the case. Interaction between members of the investigation team influences the interpretation of the findings. The team identifies new suspected locations in light of others. This step is modeled as follows [39]:

$$X'_{A_{ij}} = X_{A_{ij}} + rand_{ij}\left(X_{A_{ij}} - \left(X_{A_{kj}} + X_{A_{hj}}\right)/2\right);$$
$$i = 1,2,\cdots,NP \ \text{ and } \ j = 1,2,\cdots,D. \tag{2}$$

where i, k, and h are three possible locations of the suspect, where k and h are chosen randomly; $rand$ is a random number in the range $[-1,1]$; and X'_{A_i}-s are newly identified suspected locations ($i = 1,2,\cdots,NP$).

The team investigates newly identified suspected locations and compares them with the current ones. The current suspected locations are replaced with the corresponding new ones via a greedy manner so that the location where the suspect is more likely to

be hiding is preferred. Furthermore, the investigation team determines the best-so-far suspected location (X_{Best}) for further investigations. This can be written as:

$$X_{A_i} = X'_{A_i} \quad \text{if} \quad P'_{A_i} > P_{A_i};\ i = 1, 2, \cdots, NP. \tag{3}$$

$$X_{Best} = X'_{A_i} \quad \text{if} \quad P'_{A_i} > P_{Best};\ i = 1, 2, \cdots, NP. \tag{4}$$

where X_{Best} represents the best-so-far suspected location; P_{Best} is the probability of locating the suspect at X_{Best}; P_{A_i} is the probability of locating the suspect at X_{A_i} (representing the objective function of solution X_{A_i}). In a similar way, P'_{A_i} is defined as the probability of locating the suspect at X'_{A_i}.

The outputs of this step are X_{Best} and X_A. To decide which step to take next, the most and least probabilities of locating the suspect (P_{max} and P_{min}) are determined and compared. If $P_{min} \neq P_{max}$, the algorithm goes to step A_2; otherwise, the algorithm goes to step B_1.

2.2.1.2. Directions of inquiry (A_2). In this step of the investigation process, the investigation team suggests several directions of inquiry based on the outcome of the previous step. The team discusses the motives for the crime and considers different scenarios. Next, the team members confront the findings with the directions of inquiry they believe, which may lead to constructing new directions of inquiry or a change or confirmation of the existing directions of inquiry. This phase can be expressed as follows [39]:

$$X'_{A_{ij}} = \begin{cases} X_{Best_j} + X_{A_{dj}} + rand_{ij}\left(X_{A_{kj}} - X_{A_{hj}}\right) & \text{if} \quad rand > Pr(X_{A_i}) \\ X_{A_{ij}} & \text{otherwise} \end{cases}. \tag{5}$$

where d, k, h, and i are four possible locations of the suspect, where d, k, and h are chosen randomly; *rand* is a random number between 0 and 1; and $Pr(X_{A_i})$ is the relative probability of locating the suspect at X_{A_i}. The variables i and j are defined same as those in Eq. (1).

It can be seen from Eq. (5) that a suspected location is updated only if it has a relative probability greater than *rand*; otherwise, it remains unchanged. The relative probability of locating the suspect at X_{A_i} can be calculated as [39]:

$$Pr(X_{A_i}) = \frac{P_{A_i} - P_{min}}{P_{max} - P_{min}};\ i = 1, 2, \cdots, NP. \tag{6}$$

where P_{max} is the highest probability of locating the suspect (corresponding to the best objective function value); and P_{min} is the lowest probability of locating the suspect (corresponding to the worst objective function value).

Again, the current suspected locations ($X_{A_i};\ i = 1, 2, \cdots, NP$) are compared and replaced with the corresponding new ones ($X'_{A_i};\ i = 1, 2, \cdots, NP$) by using Eq. (3). The best-so-far suspected location is also updated by using Eq. (4).

2.2.2. Pursuit phase (B)

The pursuit phase also includes two steps, both of which model the last step of the investigation process, i.e. actions. The members of the pursuit team obey the commands of their headquarters and report back to them.

Throughout the investigation process, the investigation and pursuit teams need to work closely together to resolve complicated cases. For this purpose, the investigation team directs the pursuit team towards the best-so-far suspected location. Reciprocally, members of the pursuit team frequently report their activities to the investigation team. This interactive collaboration allows the teams to constantly update their information and maximize their efficiency. In the next two subsections, the pursuit phase is modeled.

2.2.2.1. Actions (B_1). The investigation team monitors the best-so-far suspected location and reports it to the pursuit team. All members of the pursuit team surround the designated location to arrest the suspect. This step can be mathematically expressed as follows [39]:

$$X'_{B_{ij}} = rand1_{ij}\left(X_{B_{ij}}\right) + rand2_{ij}\left(X_{Best_j} - X_{B_{ij}}\right);$$
$$i = 1, 2, \cdots, NP \ \text{ and } \ j = 1, 2, \cdots, D. \tag{7}$$

where *rand*1 and *rand*2 are two random numbers in the range $[0, 1]$; X_{Best} denotes the best-so-far suspected location (reported by the investigation team); X_{B_i} is the current position of i-th member of the pursuit team; and X'_{B_i} is the new position of i-th member of the pursuit team.

Similar to steps A_1 and A_2 in investigation phase, the current position of members of the pursuit team ($X_{B_i};\ i = 1, 2, \cdots, NP$) are compared and replaced with their corresponding new ones ($X'_{B_i};\ i = 1, 2, \cdots, NP$), and X_{Best} is updated as follows:

$$X_{B_i} = X'_{B_i} \quad \text{if} \quad P'_{B_i} > P_{B_i};\ i = 1, 2, \cdots, NP. \tag{8}$$

$$X_{Best} = X'_{B_i} \quad \text{if} \quad P'_{B_i} > P_{Best};\ i = 1, 2, \cdots, NP. \tag{9}$$

where P_{B_i} and P'_{B_i} denote the probability of locating the suspect at X_{B_i} and X'_{B_i}, respectively.

2.2.2.2. Extending the process of actions (B_2). This step completes the investigation process. The headquarters constantly update the best-so-far suspected location. Reciprocally, members of the pursuit team constantly report back their activities to the headquarters. Interaction between members of the pursuit team might be helpful to decide about the directions of inquiry and actions. The new position of members of the pursuit team will be calculated by:

$$X'_{B_{ij}} = \begin{cases} X_{B_{rj}} + rand3_{ij}\left(X_{B_{rj}} - X_{B_{ij}}\right) + rand4_{ij}\left(X_{Best_j} - X_{B_{rj}}\right) & \text{if} \quad P_{B_r} > P_{B_i} \\ X_{B_{ij}} + rand3_{ij}\left(X_{B_{ij}} - X_{B_{rj}}\right) + rand4_{ij}\left(X_{Best_j} - X_{B_{ij}}\right) & \text{if} \quad P_{B_i} > P_{B_r} \end{cases}. \tag{10}$$

where i and r are two members of the pursuit team, where r is chosen randomly; *rand*3 and *rand*4 are two random numbers in the range $[0, 1]$; X_{B_i} is the current position of i-th member of the pursuit team; X_{B_i} and X_{B_r} are the current position of i-th and r-th members of the pursuit team, respectively; and X_{B_i} is the new position of i-th member of the pursuit team.

Once again, the current position of members of the pursuit team are compared and replaced with the corresponding new ones via a greedy manner, and X_{Best} is updated (see Eqs. (8) and (9)).

2.2.3. Termination of FBI

In the last step of the algorithm, if the termination condition is satisfied, the algorithm terminates and reports the best best-so-far suspected location; otherwise, it returns to step A_1 (interpretation of the findings) to identify more new suspected locations. In this paper, the maximum number of iterations (*MaxNITs*) is considered as the termination condition of the algorithm. However, other criteria such as the maximum number of function evaluations (*MaxNFEs*) can also be considered.

The flowchart and pseudo-code of the FBI algorithm are provided in Figs. 3 and 4, respectively.

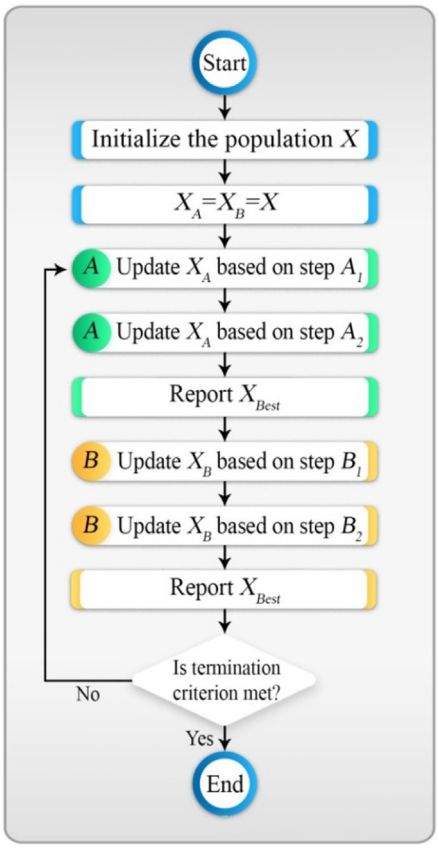

Fig. 3. Flowchart of the FBI algorithm.

Pseudo-code of Forensic-Based Investigation

Initialization phase

Set the algorithm parameters: NP and $MaxNITs$
Initialize the random population (X) by Eq. (1)
Evaluate the initial population
Determine X_{Best}
$X_A = X_B = X$
$NITs = 0$

Cyclic body of the algorithm

while $NITs < MaxNITs$
  Investigation phase (A_1)
  Identify new suspected locations (X'_A) using Eq. (2)
  Evaluate new suspected locations
  Update X_A and X_{Best} based on Eqs. (3) and (4)
  Identify the best and worst suspected locations (P_{max} and P_{min})
    Investigation phase (A_2)
    if $P_{min} \neq P_{max}$
      Identify new suspected locations (X'_A) by Eqs. (5) and (6)
      Evaluate new suspected locations
      Update X_A and X_{Best} using Eqs. (3) and (4)
    end if
  Pursuing phase (B_1)
  Generate new positions (X'_B) based on Eq. (7)
  Evaluate new positions
  Update X_B and X_{Best} by Eqs. (8) and (9)
  Pursuing phase (B_2)
  Generate new positions (X'_B) using Eq. (10)
  Evaluate new positions
  Update X_B and X_{Best} based on Eqs. (8) and (9)
  $NITs = NITs + 1$
end while
return X_{Best}

Fig. 4. Pseudo-code of the FBI algorithm.

3. Enhanced Forensic-Based investigation (EFBI)

As seen from Fig. 4, in the initialization phase of FBI, both the initial suspected locations (X_A) and the initial position of members of the pursuit team (X_B) are set to be same as the initial population (X). As a result, before the main body of the FBI algorithm starts to work, the populations X_A and X_B are similar to each other. Throughout the first phase of FBI (steps A_1 and A_2), the suspected locations ($X_{A_i}; i = 1, 2, \cdots, NP$) are updated, while the position of members of the pursuit team ($X_{B_i}; i = 1, 2, \cdots, NP$) remain unchanged. On the contrary, during the second phase (steps B_1 and B_2), the population X_B is updated, while the population X_A remains unchanged. The two phases of the FBI algorithm interact with each other only through the best-so-far suspected location (X_{Best}). At the end of the investigation phase, the investigation team reports X_{Best} to the pursuit team for further actions. Reciprocally, at the end of the pursuit phase, the pursuit team reports X_{Best} to the investigation team for further investigations. Nevertheless, members of the investigation team do not share their personal experiences with the pursuit team members and vice versa, and there seems to be a gap between the investigation and pursuit teams. The miscommunication between the teams may lead to various difficulties such as low convergence rate, trapping in local optima, or even lack of convergence. Furthermore, as mentioned before, the investigation and pursuit phases encourage the diversification and the intensification tasks of the FBI algorithm, respectively. Hence, the FBI algorithm would not be expected to be able to make a good balance between the diversification and the intensification of the search.

In this section, an enhanced version of the Forensic-Based Investigation algorithm, named as Enhanced Forensic-Based Investigation (EFBI), is proposed with the aim to reinforce communication between the investigation and pursuit teams. The steps of EFBI are same as those of the FBI. The main differences between FBI and EFBI are in: (1) the formulation for updating the position of members of the investigation team in step A_2; and (2) the way the algorithms assign population to X_A and X_B. The EFBI can be described as follows:

In the initialization phase, the initial population (X) is generated and the initial investigation team (X_A) is set to be same as the initial population (X). During the investigation phase (steps A_1 and A_2), investigation team (X_A) is updated. At the end of the investigation phase, the initial pursuit team (X_B) is set to be same as the updated investigation team (X_A). During the pursuit phase (steps B_1 and B_2), pursuit team (X_B) is updated. Finally, at the end of the pursuit phase, investigation team (X_A) is set to be same as updated pursuit team (X_B). The flowchart and pseudo-code of the EFBI algorithm is provided in Figs. 5 and 6, respectively.

In step A2 of the standard FBI, the position of members is updated based on Eq. (5). As seen from the equation, the positions are updated by using a linear summation of three components: (1) the best-so-far suspected location (X_{Best}); (2) a randomly chosen suspected location (X_{A_d}); and (3) the difference between the positions of two other randomly chosen suspected locations (X_{A_k} and X_{A_h}). It should be mentioned that there is no coefficient before the first and second components (X_{Best} and X_{A_d}), which means the coefficient is equivalent to one. As a result, new positions may become too large. So, they may exceed search space boundaries. In order to address this issue, we propose a new position update rule for step A2 of the EFBI algorithm, as follows:

$$X'_{A_{ij}} = \begin{cases} X_{Best_j} + rand_{ij}\left(X_{A_{ij}} - \left(X_{A_{kj}} + X_{A_{hj}} + X_{A_{dj}}\right)/3\right) & \text{if} \quad rand > Pr\left(X_{A_i}\right) \\ X_{A_{ij}} & \text{otherwise} \end{cases}. \tag{11}$$

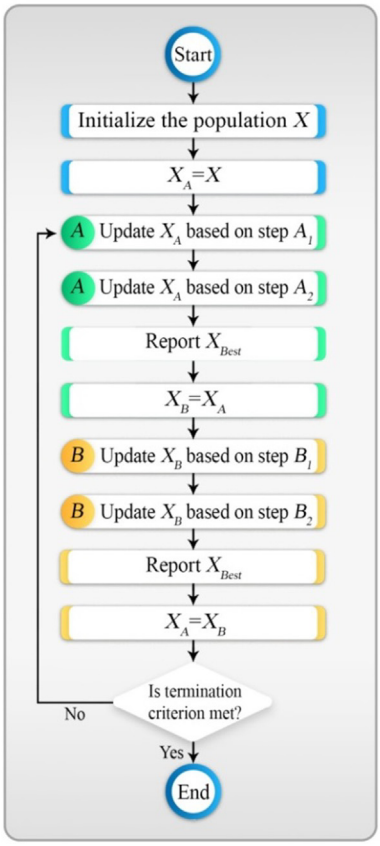

Fig. 5. Flowchart of the EFBI algorithm.

Pseudo-code of Enhanced Forensic-Based Investigation

Initialization phase

Set the algorithm parameters: NP and $MaxNITs$
Initialize the random population (X) by Eq. (1)
Evaluate the initial population
Determine X_{Best}
$X_A = X$
$NITs = 0$

Cyclic body of the algorithm

while $NITs < MaxNITs$
  Investigation phase (A_1)
  Identify new suspected locations (X'_A) using Eq. (2)
  Evaluate new suspected locations
  Update X_A and X_{Best} based on Eqs. (3) and (4)
  Identify the best and worst suspected locations (P_{max} and P_{min})
    Investigation phase (A_2)
    if $P_{min} \neq P_{max}$
      Identify new suspected locations (X'_A) by Eqs. (6) and (11)
      Evaluate new suspected locations
      Update X_A and X_{Best} using Eqs. (3) and (4)
    end if
  $X_B = X_A$
  Pursuing phase (B_1)
  Generate new positions (X'_B) based on Eq. (7)
  Evaluate new positions
  Update X_B and X_{Best} by Eqs. (8) and (9)
  Pursuing phase (B_2)
  Generate new positions (X'_B) using Eq. (10)
  Evaluate new positions
  Update X_B and X_{Best} based on Eqs. (8) and (9)
  $X_A = X_B$
  $NITs = NITs + 1$
end while
return X_{Best}

Fig. 6. Pseudo-code of the EFBI algorithm.

where d, k, h, and i are four possible locations of the suspect, where d, k, and h are chosen randomly; *rand* is a random number between 0 and 1; X_{A_i} and X'_{A_i} are the current and new positions of the i-th suspected location, respectively; X_{Best} is the best-so-far suspected location; and $Pr(X_{A_i})$ is the relative probability of locating the suspect at X_{A_i}. The variables i and j are defined same as those in Eq. (1). Fig. 7 illustrates mechanism of position update in step A2 of the EFBI algorithm.

4. Formulation of optimization problem

In this paper, the FBI and EFBI algorithms are employed for size and shape optimization of frequency-constrained truss structures. The objective is to minimize the weight of the truss structure while satisfying some constraints on natural frequencies. The mathematical formulation of the problem can be defined mathematically as [13]:

$$\text{Find}\{X\} = [x_1, x_2, \cdots, x_{nDV}]. \tag{12}$$

$$\text{to minimize} P(\{X\}) = f(\{X\}) \times f_{penalty}(\{X\}). \tag{13}$$

$$\text{subject to}: \begin{cases} \omega_j \geq \omega_j^* & \text{for some natural frequencies } j \\ \omega_k \leq \omega_k^* & \text{for some natural frequencies } k \\ x_i^L \leq x_i \leq x_i^U & i = 1, 2, \cdots, nDV \end{cases}. \tag{14}$$

where $\{X\}$ represents a solution vector containing the set of design variables; nDV is the number of design variables; x_i is the i-th design variable; $f(\{X\})$ is the objective function, which is considered as the weight of the structure in a weight minimization problem; $f_{penalty}(\{X\})$ is the penalty function, which is used to handle the problem constraints; $P(\{X\})$ is the penalized objective function to be minimized; x_i^L and x_i^U denote the lower and upper bounds of the i-th design variable, respectively; ω_j and ω_j^* are the j-th natural frequency of the structure and its corresponding lower bound, respectively; and ω_k and ω_k^* are the k-th natural frequency of the structure and its corresponding upper bound, respectively.

The objective function is the weight of the truss structure, which can be calculated as follows:

$$W(\{X\}) = \sum_{i=1}^{nE} \rho_i \times A_i \times L_i. \tag{15}$$

where $W(\{X\})$ represent the weight of the truss structure; nE is the number of structural members; ρ_i is the material density; A_i is the cross-sectional area; and L_i is the length of the i-th structural member.

Penalty functions are used in order to transform a constrained optimization problem into an unconstrained one. Here, a dynamic penalty function is employed as follows [8,41]:

$$f_{penalty}(X) = (1 + \varepsilon_1 \times \upsilon)^{\varepsilon_2}; \upsilon = \sum_{i=1}^{nC} \upsilon_i. \tag{16}$$

where nC is the number of frequency constraints; υ denotes the sum of the violations of the problem constraints; and ε_1 and ε_2 are parameters that determine the behavior of penalty function. The values of υ_i $(i = 1, 2, \cdots, nC)$ are set to zero for satisfied constraints, while in case of violated constraints, they are selected considering the severity of violation. This can be expressed as follows:

$$\upsilon_i = \begin{cases} \left|1 - \frac{\omega_i}{\omega_i^*}\right| & \text{if the } i-\text{th frequency constraint is violated} \\ 0 & \text{otherwise} \end{cases}. \tag{17}$$

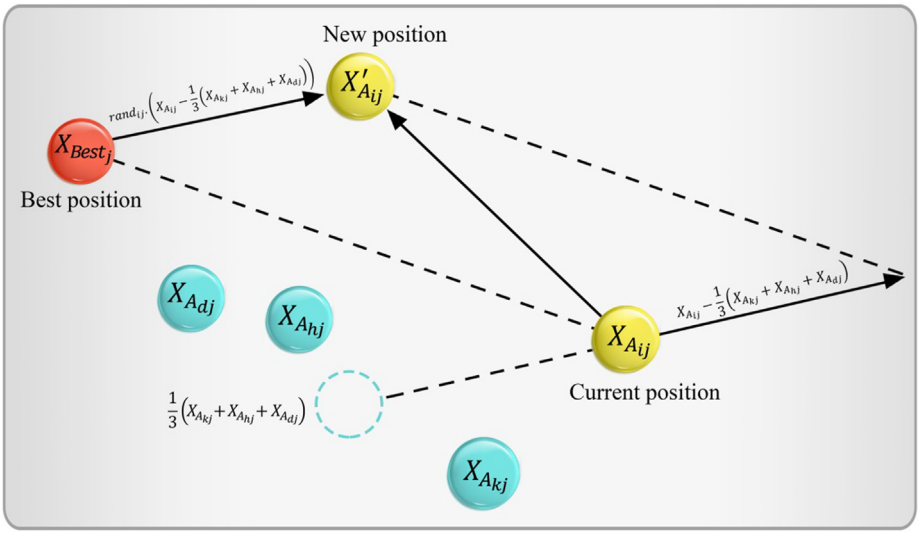

Fig. 7. New position update rule for step A_2 of the EFBI algorithm.

The values for parameters ε_1 and ε_2 determine how much a violated solution is penalized. Therefore, these parameters can affect the exploration and the exploitation of the search and can play a leading role in controlling the convergence rate [42]. So, the parameters ε_1 and ε_2 should be selected in such a way that a low degree of penalty is imposed in the early stages of the optimization process, but as the number of iterations increases, so does the degree of penalty. In other words, the severity of penalizing infeasible solutions increases as the optimization process proceeds. Such a dynamic strategy allows a better exploration of the whole search space (i.e., more diversification) in the early stages of optimization, but more emphasis on exploitation of the best solutions found (i.e., more intensification) in the final stages [11]. This helps to balance the exploration and exploitation of the search process. In this study, the parameter ε_1 is set equal to unity, while the parameter ε_2 is subjected to monotonic increase from 2 to 4.

To determine the natural frequencies and modes of vibration of a system without damping, we need to solve the following algebraic equation, which is known as the matrix eigenvalue problem [43]:

$$k\phi_n = \omega_n^2 m\phi_n. \tag{18}$$

where k and m denote the stiffness and mass matrices of the system, respectively; ω_n is the n-th natural frequency of vibration of the system $(1 = 1, 2, \cdots, N)$; and ϕ_n is the n-th natural mode of vibration of the system. N is the number of degrees of freedom of the system.

5. Case studies

In this section, three dome-like truss optimization examples are investigated to demonstrate the efficiency and effectiveness of EFBI for solving frequency-constrained truss optimization problems. The considered examples are well-known benchmark problems. The first example is size and shape optimization of a small-scale dome-like truss, while the last two ones are mid-scale and large scale dome-like truss, respectively, which are employed to show robustness of the proposed algorithm. Table 1 lists the material properties, cross-sectional area bounds, and frequency constraints of all examples. Statistical results are presented in terms of the best weight, the worst weight, average weight, standard deviation, and the maximum number of finite element analyses. It should be noted that for each example, only optimal design of the best run is reported to compare with those of other optimization algorithms. Based on the suggestion of Chou and Nguyen [39], a population size of at least 50 individuals is required to explore the search space effectively. However, our experiment results indicate that smaller population sizes result in much better performance for the investigated examples. The population size (*NP*) is set to 10 for all the problems of this study. The maximum number of structural analyses (*MaxNFEs*) is set to 10,000 for the 52-bar dome, 5000 for the 120-bar dome, and 12,000 for the 600-bar dome problems. Due to the stochastic nature of the optimization process, 10 independent runs are carried out for each algorithm on each problem. The proposed EFBI algorithm, as well as finite element analysis code, is implemented by MATLAB software.

5.1. A 52-bar dome-like truss

Simultaneous size and shape optimization of a 52-bar dome-like truss is considered as the first example. Thus, both cross sectional area of the members and nodal coordinates are considered as design variables. The initial layout of the structure is depicted in Fig. 8. All free nodes of the dome are allowed to move ±2m from their initial position in a symmetrical manner. Thus, this is a configuration optimization problem with 13 design variables (eight sizing design variables + five shape variables). As shown in Table 2, the structural members are categorized into eight different groups regarding the symmetry of the structure. Material properties, frequency constraints, and cross-sectional area bounds are provided

Table 1
Material properties, cross-sectional area bounds, and frequency constraints of various problems.

Problem	Elasticity modulus E (N/m^2)	Material density ρ (kg/m^3)	Cross-sectional area bounds (m^2)	Frequency constraints (Hz)
52-bar dome-like truss	2.1×10^{11}	7800	$0.0001 \le A_i \le 0.001$	$\omega_1 \le 50/\pi$, $\omega_2 \ge 90/\pi$
120-bar dome-like truss	2.1×10^{11}	7971.81	$0.0001 \le A_i \le 0.01293$	$\omega_1 \ge 9$, $\omega_2 \ge 11$
600-bar dome-like truss	2×10^{11}	7850	$0.0001 \le A_i \le 0.01$	$\omega_1 \ge 5$, $\omega_3 \ge 7$

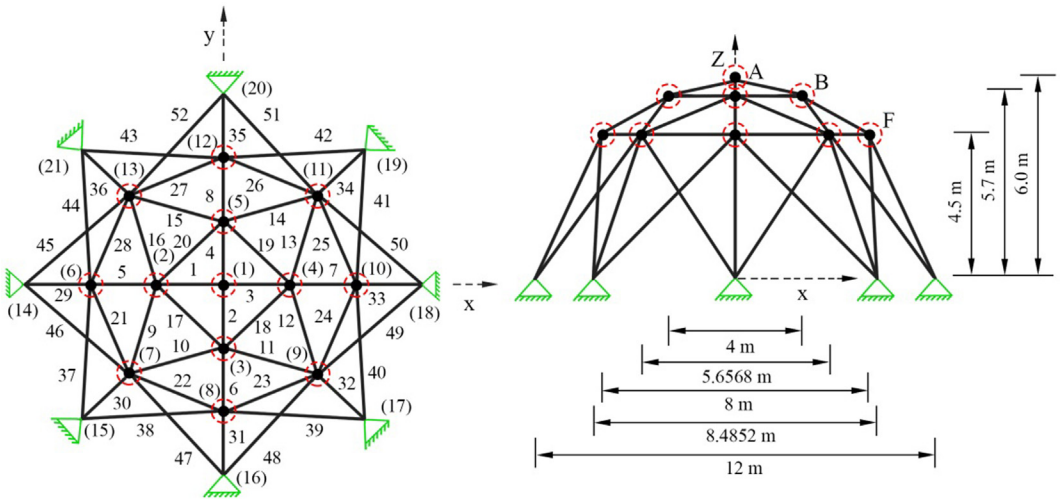

Fig. 8. Initial layout of the 52-bar dome-like truss: (a) top view, (b) side view.

Table 2
Comparison of optimal results of the 52-bar dome-like truss obtained by different algorithms.

Design variable Z_i, X_i (m); A_j (cm^2)	HS [44]	FA [44]	CSS-BBBC [8]	DPSO [45]	DE [46]	IDE [46]	ALC-PSO [47]	HALC-PSO [47]	This study	
									FBI	EFBI
Z_A	4.7374	6.4332	5.331	6.1123	6.0136	6.0052	5.6972	5.9362	**6.2948**	**6.0445**
X_B	1.5643	2.2208	2.134	2.2343	2.2802	2.3004	2.0008	2.2416	**2.2992**	**2.2002**
Z_B	3.7413	3.9202	3.719	3.8321	3.7488	3.7332	3.7000	3.7309	**4.0891**	**4.0032**
X_F	3.4882	4.0296	3.935	4.0316	3.9980	4.0000	3.8052	3.9630	**3.9059**	**3.8088**
Z_F	2.6274	2.5200	2.500	2.5036	2.5000	2.5000	2.5000	2.5000	**2.5126**	**2.5000**
A_{1-4}	1.0085	1.0050	1.0000	1.0001	1.0000	1.0001	1.0000	1.0001	**1.0064**	**1.0002**
A_{5-8}	1.4999	1.3823	1.3056	1.1397	1.0981	1.0875	1.395	1.1654	**1.1176**	**1.1620**
A_{9-16}	1.3948	1.2295	1.4230	1.2263	1.2132	1.2135	1.3184	1.2323	**1.1801**	**1.1992**
A_{17-20}	1.3462	1.2662	1.3851	1.3335	1.4227	1.4460	1.5027	1.4323	**1.3197**	**1.4108**
A_{21-28}	1.6776	1.4478	1.4226	1.4161	1.4217	1.4315	1.3888	1.3901	**1.4560**	**1.3945**
A_{29-36}	1.3704	1.0000	1.0000	1.0001	1.0001	1.0000	1.0000	1.0001	**1.0073**	**1.0000**
A_{37-44}	1.4137	1.5728	1.5562	1.5750	1.5770	1.5623	1.724	1.6024	**1.5528**	**1.4876**
A_{45-52}	1.9378	1.4153	1.4485	1.4357	1.3722	1.3724	1.3187	1.4131	**1.4348**	**1.4899**
Best weight (kg)	214.94	197.53	197.309	195.351	193.2481	193.2085	196.27	194.85	**195.98**	**193.60**
Average weight (kg)	229.88	212.80	–	198.71	196.0585	196.0478	207.13	196.85	**209.69**	**194.17**
Worst weight (kg)	–	–	–	–	202.4296	202.4215	–	–	**229.72**	**195.34**
Standard deviation (kg)	12.44	8.44	–	13.85	4.1708	4.1823	6.36	2.38	**9.56**	**0.48**
Number of FE analyses	20,000	1000	4000	6000	20,002	12,191	9000	7500	**10,000**	**10,000**

in Table 1. A non-structural mass of 50 kg is attached to all free nodes of the structure. To ensure fair comparison of optimization results, ten independent optimization runs are performed, and the optimal design of the best run is reported. This problem has been studied by several researchers by using various optimization methods: Miguel and Miguel [44] using Harmony Search (HS) and Firefly Algorithm (FA), Kaveh and Zolghadr [8] utilizing a hybridized CSS-BBBC algorithm, Kaveh and Zolghadr [45] employing Democratic Particle Swarm Optimization (DPSO), Ho-Huu et al. [46] by using Improved Differential Evolution (IDE), and Kaveh and Ilchi Ghazaan utilizing [47] hybridized optimization algorithms.

A comparison of optimal results of the 52-bar dome-like truss obtained by the proposed algorithm with other methods in the literature is presented in Table 2. It can be seen that EFBI performs considerably better than the standard FBI in terms of best weight, average weight, worst weight, and standard deviation. Moreover, the average weight, worst weight, and standard deviation given by the EFBI are better than those of other algorithms, which demonstrates the effectiveness and robustness of the EFBI. Note that although the best weights obtained by the DE and IDE are slightly lighter than that of the EFBI, these method demand more numbers of analyses than the EFBI to converge a global optimal design, i.e. DE with 20,002 FE analyses, and IDE with 12,191 FE analyses. Table 3 provides the first five optimal natural frequencies

Table 3
Natural frequencies (Hz) of the optimal designs for the 52-bar dome-like truss.

Frequency number	HS [44]	FA [44]	CSS-BBBC [8]	DPSO [45]	DE [46]	IDE [46]	ALC-PSO [47]	HALC-PSO [47]	This study	
									FBI	EFBI
1	12.2222	11.3119	12.987	11.315	11.5319	11.6033	10.4619	11.4339	**9.6775**	**11.2011**
2	28.6577	28.6529	28.648	28.648	28.6494	28.6481	28.6479	28.6480	**28.6479**	**28.6479**
3	28.6577	28.6529	28.679	28.648	28.6509	28.6481	28.6480	28.6480	**28.6483**	**28.6479**
4	28.6618	28.8030	28.713	28.650	28.6509	28.6490	28.7129	28.6482	**28.6508**	**28.6503**
5	30.0997	28.8030	30.262	28.688	28.6661	28.6530	28.8922	28.6848	**29.3780**	**28.6578**

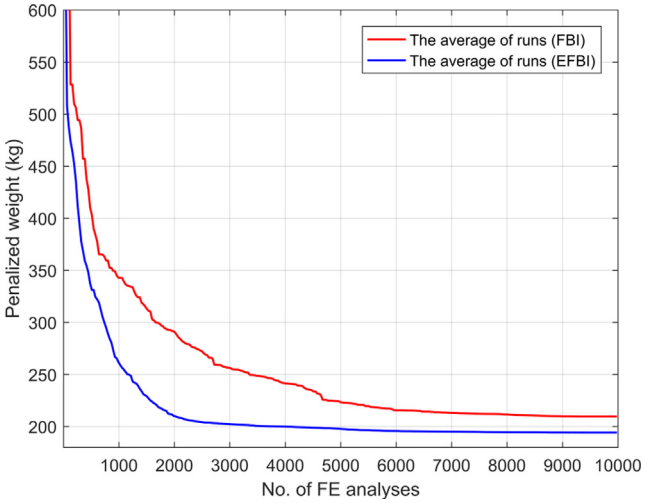

Fig. 9. Average weight convergence histories for the 52-bar dome-like truss.

obtained by different methods for the 52-bar dome-like truss. It is clear that all the natural frequencies satisfy the frequency constraints. As Table 3 suggests, constraint on the second natural frequency of the structure controls the design process. Fig. 9 presents a comparison between the average weight convergence histories of FBI and EFBI for the 52-bar dome-like truss. As can be seen from Fig. 9, EFBI has much better convergence performance compared to FBI.

5.2. A 120-bar dome-like truss

The 120-bar dome-like truss shown in Fig. 10 is considered as the second example. As shown in Fig. 10, the structural members are categorized into seven different groups considering the symmetry of the structure. The layout of the structure is kept unchanged during the optimization process. Thus, this is a size optimization problem with seven design variables. Material properties, frequency constraints, and cross-sectional area bounds are provided in Table 1. Non-structural masses are attached to all free nodes of the dome as follows: 3000 kg at node one, 500 kg at nodes 2 through 13, and 100 kg at the rest of the nodes. Ten independent optimization runs are performed, and the optimal design of the best run is reported. This problem has been solved with various methods by different researchers: Khatibinia and Naseralavi [1] using Orthogonal Multi-Gravitational Search Algorithm (OMGSA), Tejani et al. [48] employing Improved Symbiotic Organisms Search (ISOS), Kaveh and Zolghadr [45] by using Democratic Particle

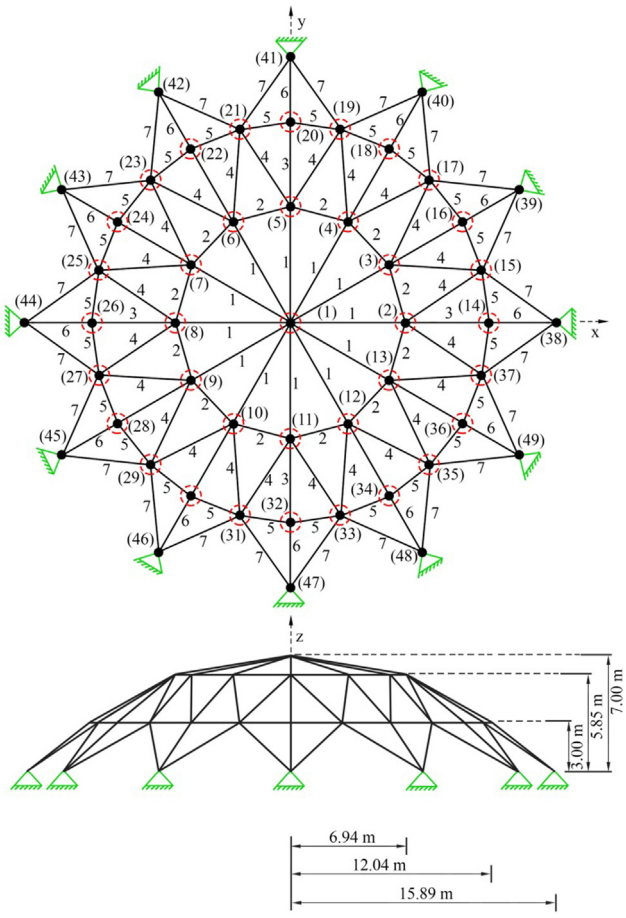

Fig. 10. Schematic of the 120-bar dome-like truss.

Swarm Optimization (DPSO), Kaveh and Zolghadr [51] utilizing Particle Swarm Ray Optimization (PSRO), Taheri and Jalili using Enhanced Biogeography-Based Optimization (EBBO), Dede et al. [50] using Jaya algorithm, and Tejani et al. [52] using a Modified Sub-Population Teaching-Learning-based Optimization (MS-TLBO).

Table 4 represents the optimal results of the 120-bar dome truss obtained by the FBI and EFBI and their results are compared to those by other studies. It is found that the EFBI gives the best results compared to the FBI and other considered methods. Similar to the previous example, the best weight, average weight, worst weight, and standard deviation obtained by the EFBI are much better than those by FBI. Table 5 provides the first five optimized natural frequencies of the best designs, which shows that all

Table 4
Comparison of optimal results of the 120-bar dome-like truss obtained by different algorithms.

Element group A_i (cm^2)	IGSA [1]	OMGSA [1]	ISOS [48]	DPSO [45]	PSRO [51]	EBBO [49]	Jaya [50]	MS-TLBO [52]	This study	
									FBI	EFBI
1	19.043	20.263	19.6662	19.607	19.972	19.8878	19.309	19.4486	**19.3805**	**19.4744**
2	41.418	39.294	39.8539	41.290	39.701	39.8248	40.763	40.3949	**40.7790**	**40.3940**
3	10.218	9.989	10.6127	11.136	11.323	10.5496	10.791	10.6921	**10.9254**	**10.6238**
4	20.664	20.563	21.2901	21.025	21.808	21.0929	21.272	21.3139	**20.9444**	**21.0395**
5	10.795	9.603	9.7911	10.060	10.179	9.4245	9.943	9.8943	**9.7849**	**9.9007**
6	12.190	11.738	11.7899	12.758	12.739	11.6648	11.695	11.7810	**11.3525**	**11.7354**
7	14.960	15.877	14.7437	15.414	14.731	15.1282	14.579	14.5979	**15.0387**	**14.9079**
Best weight (kg)	8727.28	8724.97	8710.0620	8890.48	8892.33	8711.95	8709.35	8708.729	**8710.39**	**8707.74**
Average weight (kg)	8798.55	8745.58	8728.5951	8895.99	8921.3	8718.5	8713.21	8734.7450	**8723.74**	**8710.51**
Worst weight (kg)	8800.45	8760.75	8770.8110	–	–	–	–	–	**8744.10**	**8715.18**
Standard deviation (kg)	7.195	1.183	14.2296	4.26	18.54	7.15	2.97	27.0503	**11.44**	**2.15**
Number of FE analyses	4000	4000	4000	6000	4000	6500	18,000	4000	**5000**	**5000**

Table 5
Natural frequencies (Hz) of the optimal designs for the 120-bar dome-like truss.

Frequency number	IGSA [1]	OMGSA [1]	ISOS [48]	DPSO [45]	PSRO [51]	EBBO [49]	Jaya [50]	MS-TLBO [52]	This study	
									FBI	EFBI
1	9.001	9.002	9.0001	9.0001	9.000	9.0000	9.0000	9.0002	**9.0001**	**9.0000**
2	11.003	11.003	10.9998	11.0007	11.000	11.0000	11.0002	11.0000	**11.0000**	**11.0000**
3	11.003	11.003	–	11.0053	11.005	11.0002	11.0002	11.0000	**11.0001**	**11.0000**
4	11.017	11.007	–	11.0129	11.012	11.0008	11.0008	11.0006	**11.0001**	**11.0007**
5	11.089	11.076	–	11.0471	11.045	11.0657	11.0674	11.0672	**11.0665**	**11.0679**

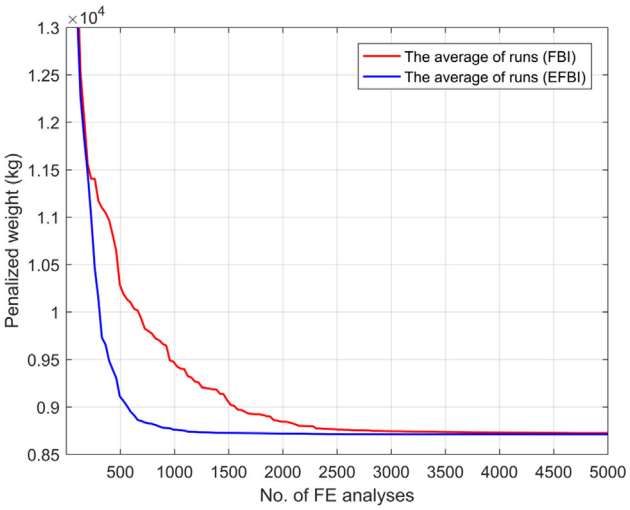

Fig. 11. Average weight convergence histories for the 120-bar dome-like truss.

frequency constraints are fulfilled. Fig. 11 shows the convergence histories of the FIB and EFBI for this example. As expected, the EFBI converges much faster than the FBI.

5.3. A 600-bar dome-like truss

Figs. 12 and 13 show the schematic of a 600-bar single-layer dome structure, which is considered as the last example. The entire

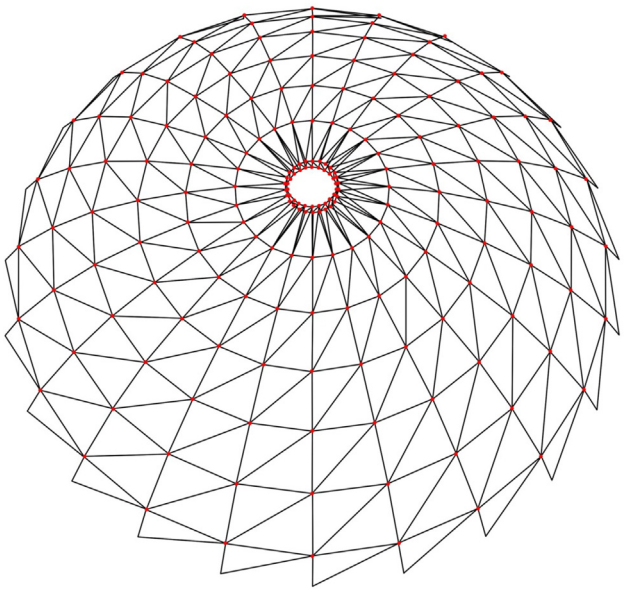

Fig. 12. Schematic of the 600-bar single-layer dome.

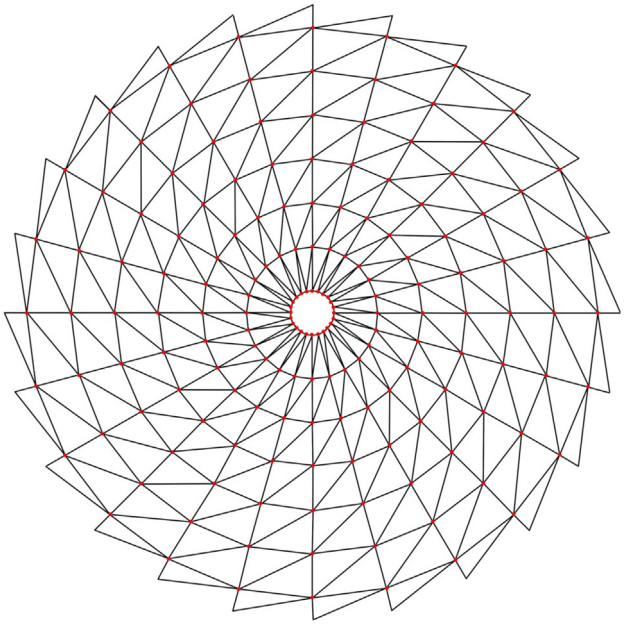

Fig. 13. The 600-bar single-layer dome-like truss (top view).

structure is composed of 216 nodes and 600 elements. The structure has a cyclically repeated pattern and could be generated by the cyclic repetition of a sub-structure composed of 9 nodes and 25 elements. Symmetric dome-like truss structures and other symmetric structures have been the subject of several group theoretic investigations mainly performed by Zingoni [53–55]. The angle of rotation of the sub-structure about the axis of revolution is 15°. The sub-structure is shown in Fig. 14 in more detail for nodal num-

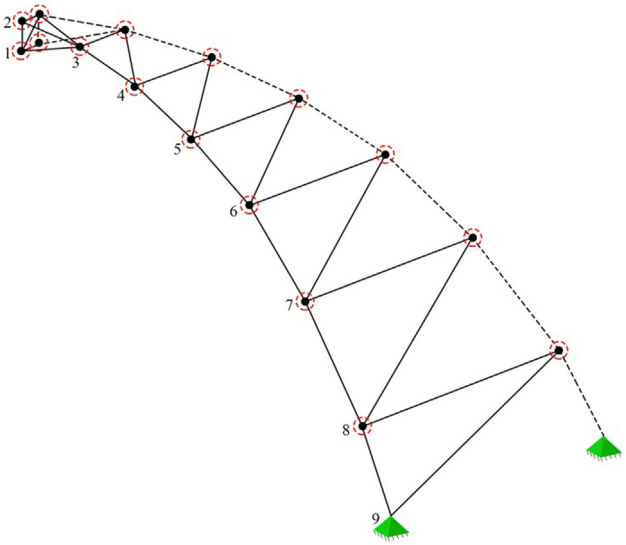

Fig. 14. Details of a sub-structure of the 600-bar single-layer dome.

Table 6
Coordinates of the nodes of the 600-bar dome-like truss.

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

bering and coordinates. Table 6 lists the nodal coordinates of the typical sub-structure in the Cartesian coordinate system. Each element of the sub-structure is considered as a design variable. The layout of the structure is kept unchanged during the optimization process. Thus, this is a size optimization problem with 25 design variables. Material density, elasticity modulus, cross-sectional area bounds, and frequency constraints of the structure are listed in Table 1. A non-structural mass of 100 kg is attached to all free nodes of the dome. This problem has been investigated by several researchers via different optimization methods: Kaveh and Zolghadr [56] using DPSO, Kaveh and Ilchi Ghazaan [57] utilizing Vibrating Particles System (VPS), Kaveh and Ilchi Ghazaan [9] employing Enhanced Colliding Bodies Optimization (ECBO) and its cascade version, Kaveh and Ilchi Ghazaan [58] using VPS and its hybrid version, and Kaveh and Ilchi Ghazaan [59] utilizing Colliding Bodies Optimization (CBO).

Table 7 provides a comparison of optimal results of the 600-bar single-layer dome obtained by the proposed algorithm with other approaches. As shown in Table 7, the best and worst weights, average weight, and standard deviation obtained by the EFBI are better than those of other algorithms. The average weight gained by EFBI is better than the best weight obtained by all the other considered methods, including the standard FBI. Even the worst weight obtained by EFBI is better than the best weight obtained by all the other methods. Furthermore, it can be seen that the EFBI requires far fewer the number of FE analysis to achieve optimal designs. A comparison of optimal results obtained by the EFBI with those of FBI reveals that the EFBI significantly performs better than the standard FBI. In fact, the best weight found by EFBI is 6076.35 kg, which is 3.12% lighter than the best weight found by FBI. Fig. 15 compares the average weight convergence histories of FBI and EFBI for the 600-bar single-layer dome. As it can be observed from

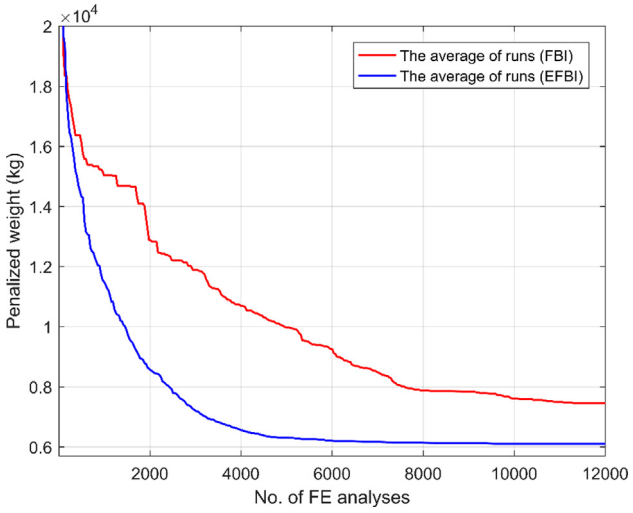

Fig. 15. Average weight convergence histories for the 600-bar dome-like truss.

Table 7
Comparison of optimal results of the 600-bar dome-like truss obtained by different algorithms (cm^2).

Element number (element nodes)	DPSO [56]	VPS [57]	ECBO [9]	ECBO-Cascade [9]	CBO [59]	VPS [58]	MDVC-UPVS [58]	This study	
								FBI	EFBI
1 (1–2)	1.365	1.3030	1.4305	1.0299	1.2404	1.3155	1.2575	**1.1565**	**1.0999**
2 (1–3)	1.391	1.3998	1.3941	1.3664	1.3797	1.2299	1.3466	**1.5495**	**1.4922**
3 (1–10)	5.686	5.1072	5.5293	5.1095	5.2597	5.5506	4.9738	**4.2881**	**6.0744**
4 (1–11)	1.511	1.3882	1.0469	1.3011	1.2658	1.3867	1.4025	**1.2744**	**1.6234**
5 (2–3)	17.711	16.9217	16.9642	17.0572	17.2255	17.4275	17.3802	**18.6386**	**17.4918**
6 (2–11)	36.266	38.1432	35.1892	34.0764	38.2991	40.1430	37.9742	**41.5551**	**37.2118**
7 (3–4)	13.263	11.8319	12.2171	13.0985	12.2234	12.8848	13.0306	**13.8532**	**12.7873**
8 (3–11)	16.919	16.6149	16.7152	15.5882	15.4712	15.5413	15.9209	**14.1519**	**14.8239**
9 (3–12)	13.333	11.3403	12.5999	12.6889	11.1577	12.2428	11.9419	**14.6455**	**12.1764**
10 (4–5)	9.534	9.3865	9.5118	10.3314	9.4636	9.3776	9.1643	**9.4627**	**9.0163**
11 (4–12)	9.884	8.7692	8.9977	8.5313	8.8250	8.6684	8.4332	**7.6139**	**8.5044**
12 (4–13)	9.547	9.6682	9.4397	9.8308	9.1021	9.1659	9.2375	**7.3169**	**8.9951**
13 (5–6)	7.866	6.9826	6.8864	7.0101	6.8417	7.1664	7.2213	**11.1706**	**7.0357**
14 (5–13)	5.529	5.4445	4.2057	5.2917	5.2882	5.2170	5.2142	**5.7036**	**5.0993**
15 (5–14)	7.007	6.3247	7.2651	6.2750	6.7702	6.5346	6.7961	**7.3517**	**6.1918**
16 (6–7)	5.462	5.1349	6.1693	5.4305	5.1402	5.4741	5.2078	**5.8131**	**4.9514**
17 (6–14)	3.853	3.3991	3.9768	3.6414	5.1827	3.6545	3.4586	**3.0348**	**3.9186**
18 (6–15)	7.432	7.7911	8.3127	7.2827	7.4781	7.6034	7.6407	**6.6133**	**7.6312**
19 (7–8)	4.261	4.4147	4.1451	4.4912	4.5646	4.2251	4.3690	**3.8692**	**4.4271**
20 (7–15)	2.253	2.2755	2.4042	1.9275	1.8617	1.9717	2.1237	**3.0077**	**2.3280**
21 (7–16)	4.337	4.9974	4.3038	4.6958	4.8797	4.5107	4.5774	**4.7851**	**4.8534**
22 (8–9)	4.028	4.0145	3.2539	3.3595	3.5065	3.5251	3.4564	**2.9061**	**3.9632**
23 (8–16)	1.954	1.8388	1.8273	1.7067	2.4546	1.9255	1.7920	**2.3371**	**1.8527**
24 (8–17)	4.709	4.7965	4.8805	4.8372	4.9128	4.7628	4.8264	**4.2659**	**4.7818**
25 (9–17)	1.410	1.5551	1.5276	2.0253	1.2324	1.6854	1.7601	**1.1603**	**1.4354**
Best weight (kg)	6344.55	6133.02	6171.51	6140.51	6182.01	6120.01	6115.10	**6272.01**	**6076.35**
Average weight (kg)	6674.71	6142.03	6191.50	6175.33	6226.37	6158.11	6119.95	**7449.82**	**6098.52**
Worst weight (kg)	–	–	–	–	–	–	–	**8852.28**	**6113.56**
Standard deviation (kg)	473.21	12.54	39.08	34.08	60.12	28.49	16.23	**855.38**	**11.95**
Number of FE analyses	9000	30,000	20,000	20,000	20,000	30,000	18,000	**12,000**	**12,000**

Table 8
Natural frequencies (Hz) of the optimal designs for the 600-bar dome-like truss.

Frequency number	DPSO [56]	VPS [57]	ECBO [9]	ECBO-Cascade [9]	CBO [59]	VPS [58]	MDVC-UPVS [58]	This study	
								FBI	EFBI
1	5.000	5.0000	5.002	5.001	5.000	5.000	5.000	**5.0001**	**5.0001**
2	5.000	5.0003	5.003	5.001	5.000	5.000	5.000	**5.0001**	**5.0001**
3	7.000	7.0000	7.001	7.001	7.000	7.000	7.000	**7.0001**	**7.0000**
4	7.000	7.0001	7.001	7.001	7.000	7.000	7.000	**7.0001**	**7.0000**
5	7.000	7.0002	7.002	7.002	7.001	7.000	7.000	**7.0003**	**7.0024**

Fig. 15, the convergence rate of EFBI is much higher than that of FBI. The first five optimal natural frequencies obtained by various methods for the 600-bar dome-like truss are presented in Table 8. The results confirm that the frequency constraints of the optimum designs obtained by FBI and EFBI are satisfied.

6. Conclusion

Forensic-Based Investigation (FBI) is a human-based metaheuristic algorithm, which has been developed recently inspired by the forensic investigation process. The major drawback of the FBI algorithm is the lack of an effective interaction between the diversification and the intensification phases of the search. In order to overcome this drawback, an enhanced FBI (EFBI) is proposed in this paper, which aims to reinforce communication between the investigation and pursuit teams (i.e., between the the diversification and the intensification tasks). The other improvement is carried out on the position update rule of the investigation step. In the investigation phase of FBI (step A_2), the position update rule (Eq. (5)) yields too large step sizes, which may result in divergence and strange oscillations and instabilities. To address this issue, we propose a new position update rule involving the best-so-far suspected location and the current suspected locations. Both the FBI and the EFBI are then applied to solve size and shape optimization problem of dome truss structures with frequency constraints. This is the first time that the FBI is applied in the field of structural optimization. Numerical results reveal that in most of the case studies, the average weight and standard deviation on average weight obtained by the EFBI are better than those gained by some other methods in the literature. Furthermore, thanks to the good balance between the diversification and the intensification tasks, the EFBI performs significantly better than the FBI in terms of efficiency and effectiveness. Especially, in all case studies, the best weight, average weight, worst weight, and standard deviation on average weight obtained by the EFBI are considerably better than those of the FBI. This shows that the EFBI can offer a high-performance optimization algorithm for solving truss optimization problems with multiple frequency constraints. Finally, the proposed algorithm could be extended to other structural optimization problems, such as frames, retaining walls, etc.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Khatibinia M, Naseralavi SS. Truss optimization on shape and sizing with frequency constraints based on orthogonal multi-gravitational search algorithm. J Sound Vib 2014;333(24):6349–69. https://doi.org/10.1016/j.jsv.2014.07.027.

[2] Grandhi R, Venkayyat VB. Structural optimization with frequency constraints. AIAA J 1988;26(7):858–66. https://doi.org/10.2514/3.9979.

[3] Bellagamba L, Yang TY. Minimum-mass truss structures with constraints on fundamental natural frequency. AIAA J 1981;19(11):1452–8. https://doi.org/10.2514/3.7875.

[4] Tong WH, Liu GR. An optimization procedure for truss structures with discrete design variables and dynamic constraints. Comput Struct 2001;79(2):155–62. https://doi.org/10.1016/S0045-7949(00)00124-3.

[5] Sedaghati R, Suleman A, Tabarrok B. Structural optimization with frequency constraints using the finite element force method. AIAA J 2002;40(2):382–8. https://doi.org/10.2514/2.1657.

[6] Lingyun W, Mei Z, Guangming W, Guang M. Truss Optimization on Shape and Sizing with Frequency Constraints Based on Genetic Algorithm. Comput Mech 2005;35(5):361–8. https://doi.org/10.1007/s00466-004-0623-8.

[7] Gomez HM. Truss optimization with dynamic constraints using a particle swarm algorithm. Expert Syst Appl 2011;38:957–68. https://doi.org/10.1016/j.eswa.2010.07.086.

[8] Kaveh A, Zolghadr A. Truss optimization with natural frequency constraints using a hybridized CSS-BBBC algorithm with trap recognition capability. Comput Struct 2012;102:14–27. https://doi.org/10.1016/j.compstruc.2012.03.016.

[9] Kaveh A, Ilchi Ghazaan M. Optimal design of dome truss structures with dynamic frequency constraints. Struct Multidiscipl Optim 2016;53(3):605–21. https://doi.org/10.1007/s00158-015-1357-2.

[10] Ho-Huu V, Nguyen-Thoi T, Truong-Khac T, Le-Anh L, Vo-Duy T. An improved differential evolution based on roulette wheel selection for shape and size optimization of truss structures with frequency constraints. Neural Comput & Applic 2018;29(1):167–85. https://doi.org/10.1007/s00521-016-2426-1.

[11] Kaveh A, Zolghadr A. Optimal design of cyclically symmetric trusses with frequency constraints using cyclical parthenogenesis algorithm. Adv Struct Eng 2018;21(5):739–55. 10.1177%2F1369433217732492.

[12] Lieu QX, Do DT, Lee J. An adaptive hybrid evolutionary firefly algorithm for shape and size optimization of truss structures with frequency constraints. Comput Struct 2018;195:99–112. https://doi.org/10.1016/j.compstruc.2017.06.016.

[13] Kaveh A, Biabani Hamedani K, Kamalinejad M. Set theoretical variants of the teaching–learning-based optimization algorithm for optimal design of truss structures with multiple frequency constraints. Acta Mech 2020;231(9):3645–72. https://doi.org/10.1007/s00707-020-02718-3.

[14] Grandhi R. Structural optimization with frequency constraints-a review. AIAA J 1993;31(12):2296–303. https://doi.org/10.2514/3.11928.

[15] Talbi EG. Metaheuristics: from design to implementation. John Wiley & Sons; 2009.

[16] Yang XS. Nature-inspired metaheuristic algorithms. Luniver press 2010.

[17] Mirjalili S, Lewis A. The whale optimization algorithm. Adv Eng Softw 2016;95:51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008.

[18] Holland JH. Genetic algorithms. Sci Am 1992;267(1):66–73.

[19] Dasgupta D, Michalewicz Z, editors. Evolutionary algorithms in engineering applications. Springer Science & Business Media; 2013.

[20] Nordin P, Keller RE, Francone FD. Genetic programming. In: Banzhaf W, editor. Springer; 1998.

[21] Fogel LJ. Intelligence through simulated evolution: forty years of evolutionary programming. John Wiley & Sons, Inc.; 1999.

[22] Moscato P. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. Caltech concurrent computation program. C3P Report 1989;1989:826.

[23] Storn R, Price K. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. J Global Optim 1997;11(4):341–59. https://doi.org/10.1023/A:1008202821328.

[24] Rashedi E, Nezamabadi-Pour H, Saryazdi S. GSA: a gravitational search algorithm. Inf Sci 2009;179(13):2232–48. https://doi.org/10.1016/j.ins.2009.03.004.

[25] Genç HM, Eksin I, Erol OK. Big Bang-Big Crunch optimization algorithm hybridized with local directional moves and application to target motion analysis problem. In: 2010 IEEE International Conference on Systems, Man and Cybernetics 2010 Oct 10. IEEE, pp. 881-887. https://doi.org/10.1109/ICSMC.2010.5641871

[26] Kaveh A, Talatahari S. A novel heuristic optimization method: charged system search. Acta Mech 2010;213(3):267–89. https://doi.org/10.1007/s00707-009-0270-4.

[27] Kaveh A, Khayatazad M. A new meta-heuristic method: ray optimization. Comput Struct 2012;112(3):283–94. https://doi.org/10.1016/j.compstruc.2012.09.003.
[28] Eskandar H, Sadollah A, Bahreininejad A, Hamdi M. Water cycle algorithm–A novel metaheuristic optimization method for solving constrained engineering optimization problems. Comput Struct 2012;110:151–66. https://doi.org/10.1016/j.compstruc.2012.07.010.
[29] Karaboga D. An idea based on honey bee swarm for numerical optimization. Technical report-tr06, Erciyes University, engineering faculty, computer engineering department; 2005 Oct.
[30] Kennedy J, Eberhart R. Particle swarm optimization. In Proceedings of ICNN'95-International Conference on Neural Networks 1995 Nov 27, Vol. 4. IEEE, pp. 1942-1948. https://doi.org/10.1109/ICNN.1995.488968
[31] Dorigo M, Birattari M, Stutzle T. Ant colony optimization. IEEE Comput Intell Mag 2006;1(4):28–39. https://doi.org/10.1109/MCI.2006.329691.
[32] Yang XS, Deb S. Cuckoo search via Lévy flights. In 2009 World congress on nature & biologically inspired computing (NaBIC) 2009 Dec 9. IEEE, pp. 210-214. https://doi.org/10.1109/NABIC.2009.5393690
[33] Geem ZW, Kim JH, Loganathan GV. A new heuristic optimization algorithm: harmony search. Simulation 2001;76(2):60–8. 10.1177%2F003754970107600201.
[34] Kashan AH. League championship algorithm: a new algorithm for numerical function optimization. In: 2009 international conference of soft computing and pattern recognition 2009 Dec 4. IEEE, pp. 43-48. https://doi.org/10.1109/SoCPaR.2009.21
[35] Rao RV, Savsani VJ, Vakharia DP. Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput Aided Des 2011;43(3):303–15. https://doi.org/10.1016/j.cad.2010.12.015.
[36] Atashpaz-Gargari E, Lucas C. Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In: 2007 IEEE congress on evolutionary computation 2007 Sep 25. IEEE, pp. 4661–4667. https://doi.org/10.1109/CEC.2007.4425083
[37] Wagan AI, Shaikh MM. A new metaheuristic optimization algorithm inspired by human dynasties with an application to the wind turbine micrositing problem. Appl Soft Comput 2020;90:. https://doi.org/10.1016/j.asoc.2020.106176106176.
[38] Gandomi AH. Interior search algorithm (ISA): a novel approach for global optimization. ISA Trans 2014;53(4):1168–83. https://doi.org/10.1016/j.isatra.2014.03.018.
[39] Chou JS, Nguyen NM. FBI inspired meta-optimization. Appl Soft Comput 2020;93:. https://doi.org/10.1016/j.asoc.2020.106339106339.
[40] Salet R. Framing in criminal investigation: How police officers (re) construct a crime. Police J 2017;90(2):128–42. 10.1177%2F0032258X16672470.
[41] Kaveh A. Advances in metaheuristic algorithms for optimal design of structures. 3rd ed. Switzerland: Springer International Publishing; 2021.
[42] Joines JA, Houck CR. On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GA's. In: Proceedings of the first IEEE conference on evolutionary computation. IEEE world congress on computational intelligence 1994 Jun 27. IEEE, pp. 579-584. https://doi.org/10.1109/ICEC.1994.349995
[43] Chopra AK. Dynamics of structures. Upper Saddle River, NJ: Pearson Education; 2012.
[44] Miguel LF, Miguel LF. Shape and size optimization of truss structures considering dynamic constraints through modern metaheuristic algorithms. Expert Syst Appl 2012;39(10):9458–67. https://doi.org/10.1016/j.eswa.2012.02.113.
[45] Kaveh A, Zolghadr A. Democratic PSO for truss layout and size optimization with frequency constraints. Comput Struct 2014;130:10–21. https://doi.org/10.1016/j.compstruc.2013.09.002.
[46] Ho-Huu V, Vo-Duy T, Luu-Van T, Le-Anh L, Nguyen-Thoi T. Optimal design of truss structures with frequency constraints using improved differential evolution algorithm based on an adaptive mutation scheme. Autom Construct 2016;68:81–94. https://doi.org/10.1016/j.autcon.2016.05.004.
[47] Kaveh A, Ilchi Ghazaan M. Hybridized optimization algorithms for design of trusses with multiple natural frequency constraints. Adv Eng Softw 2015;79:137–47. https://doi.org/10.1016/j.advengsoft.2014.10.001.
[48] Tejani GG, Savsani VJ, Patel VK, Mirjalili S. Truss optimization with natural frequency bounds using improved symbiotic organisms search. Knowl Based Syst 2018;143:162–78. https://doi.org/10.1016/j.knosys.2017.12.012.
[49] Taheri SHS, Jalili S. Enhanced biogeography-based optimization: a new method for size and shape optimization of truss structures with natural frequency constraints. Lat Am J Solids Stru 2016;13(7):1406–30. https://doi.org/10.1590/1679-78252208.
[50] Dede T, Grzywiński M, Rao RV. Jaya: A New Meta-heuristic Algorithm for the Optimization of Braced Dome Structures. In: Advanced Engineering Optimization Through Intelligent Techniques 2020. Springer, Singapore, pp. 13–20. https://doi.org/10.1007/978-981-13-8196-6_2.
[51] Kaveh A, Zolghadr A. A new PSRO algorithm for frequency constraint truss shape and size optimization. Struct Eng Mech 2014;52(3):445–68. https://doi.org/10.12989/sem.2014.52.3.445.
[52] Tejani GG, Savsani VJ, Patel VK. Modified sub-population teaching-learning-based optimization for design of truss structures with natural frequency constraints. Mech Based Des Struct Mach 2016;44(4):495–513. https://doi.org/10.1080/15397734.2015.1124023.
[53] Zingoni A. Symmetry recognition in group-theoretic computational schemes for complex structural systems. Comput Struct 2012;94:34–44. https://doi.org/10.1016/j.compstruc.2011.12.004.
[54] Zingoni A. On the best choice of symmetry group for group-theoretic computational schemes in solid and structural mechanics. Comput Struct 2019;223:. https://doi.org/10.1016/j.compstruc.2019.106101106101.
[55] Zingoni A. Use of symmetry groups for generation of complex space grids and group-theoretic vibration analysis of triple-layer grids. Eng Struct 2020;223. https://doi.org/10.1016/j.engstruct.2020.111177.
[56] Kaveh A. Optimal analysis and design of large-scale domes with frequency constraints. In: Applications of Metaheuristic Optimization Algorithms in Civil Engineering 2017. Springer, Cham, pp. 257–279. https://doi.org/10.1007/978-3-319-48012-1_14
[57] Kaveh A, Ghazaan Ilchi M. Vibrating particles system algorithm for truss optimization with multiple natural frequency constraints. Acta Mech 2017;228(1):307–22. https://doi.org/10.1007/s00707-016-1725-z.
[58] Kaveh A, Ilchi Ghazaan M. A new hybrid meta-heuristic algorithm for optimal design of large-scale dome structures. Eng Optim 2018;50(2):235–52. https://doi.org/10.1080/0305215X.2017.1313250.
[59] Kaveh A, Ilchi Ghazaan M. Meta-heuristic algorithms for optimal design of real-size structures. Switzerland: Springer International Publishing; 2018.