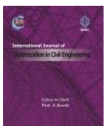
INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING
Int. J. Optim. Civil Eng., 2022; 12(3):335-364



DISCRETE AND CONTINUOUS SIZING OPTIMIZATION OF LARGE-SCALE TRUSS STRUCTURES USING DE-MEDT ALGORITHM

A. Kaveh[*, †] and S. M. Hosseini
School of Civil Engineering, Iran University of Science and Technology, P.O. Box 16846-13114, Iran

ABSTRACT

Design optimization of structures with discrete and continuous search spaces is a complex optimization problem with lots of local optima. Metaheuristic optimization algorithms, due to not requiring gradient information of the objective function, are efficient tools for solving these problems at a reasonable computational time. In this paper, the Doppler Effect-Mean Euclidian Distance Threshold (DE-MEDT) metaheuristic algorithm is applied to solve the discrete and continuous optimization problems of the truss structures subject to multiple loading conditions and design constraints. DE-MEDT algorithm is a recently proposed metaheuristic developed based on a physical phenomenon called Doppler Effect (DE) with some idealized rules and a mechanism called Mean Euclidian Distance Threshold (MEDT). The efficiency of the DE-MEDT algorithm is evaluated by optimizing five large-scale truss structures with continuous and discrete variables. Comparing the results found by the DE-MEDT algorithm with those of other existing metaheuristics reveals that the DE-MEDT optimizer is a suitable optimization technique for discrete and continuous design optimization of large-scale truss structures.

Keywords: metaheuristics; Doppler effect; mean Euclidian distance threshold mechanism; discrete optimization; continuous optimization; truss structures.

Received: 20 February 2022; Accepted: 21 April 2022

1. INTRODUCTION

Discrete and continuous optimization of structures is known as a complex optimization

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

problem with many local optima. In this problem, optimization aims to minimize the weight of the structures subject to multiple loading conditions and design constraints [1]. Design constraints are the stress and displacement requirements, and design variables are the cross-sectional areas of the structure members. These variables can be selected from a discrete set or can take a value from a continuous range. Although many researchers have investigated the weight minimization of the truss structures [2-8], it is still one of the most challenging problems to be demonstrated. The design optimization problem cannot be practically solved with traditional methods due to their limitations, such as requiring gradient information and dependency on the starting point. Metaheuristic Optimization Algorithms (MOAs), as probabilistic solvers, are free from these restrictions and solve this problem at a reasonable computational time. Since MOAs have these advantages, their developments have received considerable attention from the scientific community in recent years.

MOAs are efficient and robust solvers developed to tackle hard optimization problems. Most of them are inspired by nature. Each MOA has two conflicting search mechanisms [9]: exploration (diversification or global search) and exploitation (intensification or local search). In the exploration phase, MOA deeply explores various regions of the search space during the early steps of the searching process. The exploitation phase is usually performed after the exploration stage. In this phase, the algorithm probes the neighborhood of better-quality solutions. A good trade-off between exploration and exploration abilities leads to a well-organized MOA.

According to the source of inspiration, MOAs can generally be categorized into three main groups: Evolutionary-based, Swarm-based, and Physics-based algorithms. Evolutionary-based algorithms mimic biological evolution mechanisms such as reproduction, mutation, and selection. Genetic Algorithm (GA) [10], Evolution Strategy (ES) [11], and Differential Evolution (DE) [12] are the most well-known evolutionary-based MOAs. Swarm-based algorithms are another type of MOAs that mimic the social behavior of animals living in swarms. The most famous examples of swarm-based MOAs are Particle Swarm Optimization (PSO) [13], Artificial Bee Colony (ABC) [14], and Cuckoo Search (CS) [15]. As the last group of MOAs, Physics-based algorithms are inspired by the physical phenomena to update candidate solutions iteratively. Gravitational Search Algorithm (GSA) [16], Charged System Search (CSS) [17], and Plasma Generation Optimization (PGO) [18] are the instances of physics-based optimization algorithms.

In this paper, a newly developed population-based metaheuristic so-called Doppler Effect-Mean Euclidian Distance Threshold (DE-MEDT) algorithm is applied to the optimal design of large-scale truss structures with discrete and continuous design variables. Optimization aims to minimize the weight of the truss structures under stress and displacement constraints. DE-MEDT algorithm is a population-based metaheuristic inspired by the physical phenomenon called Doppler Effect (DE) [19]. In the cyclic body of the algorithm, a mechanism called Mean Euclidian Distance Threshold (MEDT) is also developed to improve the quality of the candidate solutions, which are called observers. MEDT mechanism also decreases the possibility of trapping into local optima. The performance of the DE-MEDT algorithm is demonstrated through five large-scale truss structures. The results found by the DE-MEDT algorithm are compared with those of some other state-of-the-art metaheuristics existing in the literature.

The remaining sections of this paper are organized as follows. In Section 2, the DE-MEDT algorithm is comprehensively explained, and its pseudo-code is presented. In Section 3, the objective of the optimization problem is mathematically stated. In the subsections of Section 3, the performance of the DE-MEDT metaheuristic algorithm is illustrated through five large-scale truss structures under stress and displacement constraints, and its results are compared with those of some other state-of-the-art metaheuristics in the literature. The concluding remarks are finally driven in Section 4.

2. DOPPLER EFFECT-MEAN EUCLIDIAN DISTANCE THRESHOLD

Doppler Effect-Mean Euclidian Distance Threshold (DE-MEDT) algorithm is a physics-based metaheuristic that has been recently proposed by Kaveh et al. [19]. The proposed DE-MEDT metaheuristic algorithm has been developed based on the Doppler Effect (DE) phenomenon and a new mechanism called Mean Euclidian Distance Threshold (MEDT). In DE-MEDT algorithm, the search agents are defined as observers, and the population size is fixed equal to the number of observers in the search space. Thus, the DE-MEDT algorithm is a population-based optimizer in which each candidate solution containing a number of optimization variables is considered as an observer. The proposed algorithm comprises three main phases to perform the optimization process. These phases include initialization, position updating of the observers based on the DE equation, and MEDT mechanism. Some idealized assumptions are considered to update the position of observers based on DE formulation. By incorporating the MEDT mechanism in the cyclic body of the algorithm, the quality of the observers is improved, and the algorithm can find promising optimum solutions. The MEDT mechanism, due to having some suitable features, decreases the possibility of being trapped in local minima and makes a good balance between the exploration and exploitation tendencies of the algorithm.

2.1 Initialization phase

Like other population-based metaheuristics, DE-MEDT starts with a set of candidate solutions called observers (X) as shown in Eq. (1), which is generated stochastically.

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & x_{1,d-1} & x_{1,d} \\ x_{2,1} & \cdots & x_{2,j} & \cdots & x_{2,d} \\ \cdots & \cdots & x_{i,j} & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{nOs-1,1} & \cdots & x_{nOs-1,j} & \cdots & x_{nOs-1,d} \\ x_{nOs,1} & \cdots & x_{nOs,j} & x_{nOs,d-1} & x_{nOs,d} \end{bmatrix} \tag{1}$$

in which X denotes a set of candidate solutions generated randomly using Eq. (2), $x_{i,j}$ represents the jth position of the ith solution, nOs is the number of candidate solutions (observers), and d is the dimension size of the problem being optimized.

$$x_{i,j} = x_{j,min} + rand \times \left(x_{j,max} - x_{j,min}\right) \;\; ; \;\; i = 1,2,\ldots,nOs, \;\; j = 1,2,..,d \tag{2}$$

where $rand$ is a random value generated between 0 and 1, $x_{j,max}$ and $x_{j,min}$ are the upper and lower bounds of the jth design variable, respectively. Each randomly-generated candidate solution is then evaluated and sorted in ascending order. Thus, the first and last members will become the initial population's best and worst candidate solutions, respectively.

2.2. Position updating of the observers based on DE equation

In the cyclic body of the algorithm, the observers update their positions based on the DE formulation with some idealized rules. The primary formulation of the DE can be stated as follows [20]:

$$f_o = f_s\left(\frac{v + v_o}{v + v_s}\right) \tag{3}$$

where f_o is the frequency perceived by the observer, f_s is the frequency of the source, and v, v_o, and v_s are respectively the velocity of the wave in a stationary medium, and the velocities of the observer and source with respect to this medium. The mathematical relationship between the wavelength and frequency is stated as follows [21]:

$$v = f\lambda \tag{4}$$

where v, f, and λ respectively denote the propagation speed in the medium (m/s), the frequency (Hz), and wavelength (m). According to this equation, the frequency and wavelength are inversely proportional to each other so that an increase in frequency leads to decreasing in wavelength and vice versa [21]. Using Eq. (4), the mathematical relationship between the wavelength and frequency can be also obtained for both observer and source as follows:

$$f_o = \frac{v}{\lambda_o} \tag{5}$$

$$f_s = \frac{v}{\lambda_s} \tag{6}$$

In the implementation of DE-MEDT, the velocities of the observer and source are used, and the effect of perceived frequency by the observer and emitted frequency by the source are eliminated from the DE equation. Considering this idealized assumption, the propagation speed in the numerator of Eqs. (5) and (6) are respectively replaced with the velocities of v_o and v_s:

$$f_o = \frac{v_o}{\lambda_o} \tag{7}$$

$$f_s = \frac{v_s}{\lambda_s} \tag{8}$$

By substituting Eqs. (5) and (6) in Eq. (1) and manipulating it, the new velocity, v_o^{new}, as stepsize for updating the position of the observers is calculated, as shown in Eq. (9). In the DE-MEDT algorithm, the new position of the ith observer (i.e., position updating equation of the observers) is calculated, which is shown in Eq. 10. Fig. 1 schematically illustrates the observers' position updating based on the DE equation.

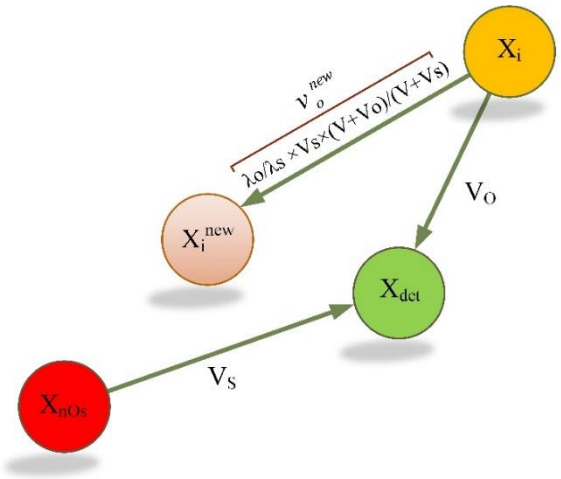


Figure 1. Position updating of the observers in the DE-MEDT algorithm

$$v_o^{new} = \frac{\lambda_o}{\lambda_s} \times v_s \times \left(\frac{v + v_o}{v + v_s}\right) \tag{9}$$

$$x_i^{new} = x_i + v_o^{new} \tag{10}$$

in which $\frac{\lambda_o}{\lambda_s}$ is considered as a random vector generated between 0 and 1. v_o, v_s, and v are respectively the velocities of the observer, source, and propagation velocity of the medium, which are calculated by the following equations:

$$v_o = x_{det} - x_i \tag{11}$$

$$v_s = x_{det} - x_{nOs} \tag{12}$$

$$v = x_{det} \tag{13}$$

where x_{det} is the position of the determinative agent. For calculating x_{det}, the observer with better quality (i.e., objective function value) than the ith observer (x_i) is selected randomly from the sorted population. The randomly selected observer is called a determinative observer (x_{det}). x_{nOs} is the position of the last observer in the sorted population. In the

sorted population, candidate solutions (observers) are arranged based on the objective function values in ascending order. Therefore, the first and last observers are the best and worst solutions of the population, respectively.

2.3 MEDT mechanism

DE-MEDT algorithm is equipped with a mechanism to afford a chance to escape from local optima and prevent probable premature convergence. Thus, a new mechanism called Mean Euclidian Distance Threshold (MEDT) is proposed in the present algorithm. This mechanism is performed after determining the new position of the ith observer by Eq. (10). The following scheme mathematically states how this mechanism performs, and Fig. 1 schematically indicates the procedure of this mechanism during the course of iterations.

$$\begin{gathered} \boldsymbol{if}\ r_1 < pa \times (1 - CI) \\ x_{i,m}^{new} = x_{1,m}^{new} \times r_2 \times SRI^{Iter} \\ \boldsymbol{end} \end{gathered} \tag{14}$$

where r_1 is a random number in the range of $[0,1]$. pa is a probability ratio determining how likely this mechanism works. CI is the convergence index indicating the convergence of the solution in the current iteration. This index can be determined by using Eq. (15). m is a random integer number, which can be generated from 1 to the number of design variables (d). $x_{i,m}^{new}$ and $x_{1,m}^{new}$ denote the position of mth design variable of the ith and the first newly generated observers, respectively. x_1^{new} is not necessarily the best observer and is just the first newly generated observer in the current iteration. r_2 is a random number in the range of $[-1,1]$. $Iter$ denotes the current iteration number. SRI^{Iter} represents the Scatter Radius Index (SRI) in the current iteration. This radius determines how much the observer's positions are ideally close to each other in the search space of the current iteration. It can be calculated by averaging all agents' Euclidian distances of the current iteration, as shown in Eq. (16). Eq. (14) indicates that if $r_1 < pa \times (1 - CI)$, one dimension of the x_i^{new} obtained by using Eq. (10) is selected, and its value is regenerated with the value equal to $x_{1,m}^{new} \times r_2 \times SRI^{Iter}$. This regeneration will wisely improve the quality of the solutions and decrease the possibility of being trapped in local optima.

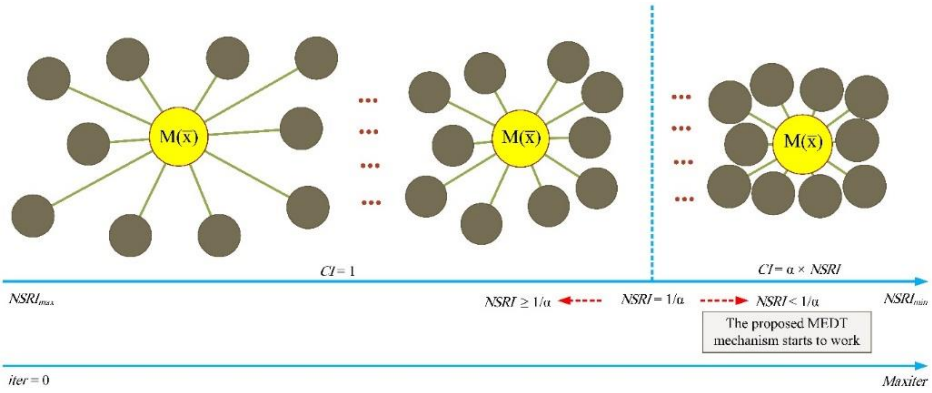


Figure 2. Schematic representation of performing MEDT mechanism during the course of iterations

$$CI = \begin{cases} \alpha \times NSRI^{Iter} & if\ NSRI^{Iter} < 1/\alpha \\ 1 & Otherwise \end{cases} \tag{15}$$

$$SRI^{Iter} = \frac{1}{nOs}\sum_{i=1}^{nOs} ED_i \tag{16}$$

where α is a sensitive parameter, determines the converge criterion of the algorithm. $NSRI^{Iter}$ is the normalized value of the SRI^{Iter}, calculates by using Eq. (17). nOs is the number of candidate solutions (i.e., observers). ED_i denotes the Euclidian distance of ith observer from the mean position of the observers, which is calculated using Eq. (18).

$$NSRI^{Iter} = \frac{SRI^{Iter}}{max(UB - LB)} \tag{17}$$

$$ED_i = \sqrt{\sum_{j=1}^{d}\left(x_{i,j} - Mean_j\right)^2} \tag{18}$$

where UB and LB denote to the lower and upper bounds of the given problem, respectively. $Mean_j$ is the average value of the jth design variable, which is calculated by Eq. (19). By calculating the average position of all design variables ($j = 1,2,..,d$), the average position of the algorithm agents ($M(\bar{x})$), as in Eq. (20), is obtained.

$$Mean_j = \frac{1}{nOs}\sum_{i=1}^{nOs} x_{i,j} \tag{19}$$

$$M(\bar{x}) = \left[Mean_1, Mean_2, \dots, Mean_j, \dots, Mean_d\right] \tag{20}$$

After position updating of each observer using Eq. (10) and (14), the design variables of the observer (i.e., x_i^{new}) are controlled to be in the range between lower and upper bounds of the given problem. If the jth design variables of the newly generated candidate solution (i.e., $x_{i,j}^{new}$) be out of the permissible range, its value will be replaced by the boundary value of the closer one, which is mathematically shown in Eqs. (21) and (22).

$$x_{i,j}^{new} = max\left(x_{j,min}, x_{i,j}^{new}\right) \tag{21}$$

$$x_{i,j}^{new} = min\left(x_{j,max}, x_{i,j}^{new}\right) \tag{22}$$

where $x_{j,min}$ and $x_{j,max}$ denote to the minimum and maximum permissible value of the jth design variable of the given problem.

2.4 Evaluating and sorting the observers

After checking the newly generated observer (x_i^{new}) to be in the permissible range using Eq. (21) and (22), it is evaluated. When all observers of the current population are evaluated, the greedy strategy is performed. According to this strategy, the observers generated in the current iteration and those created in the previous iteration are merged. Then, the merged population is sorted in ascending order, and the first nOS observers of the sorted population are selected. The selected observers are considered as the current population and will be used in the next iteration. Employing the strategy illustrates the intensification ability of the DE-MEDT algorithm.

2.5 Termination of DE-MEDT algorithm

In the last step, the termination criterion of the DE-MEDT algorithm is controlled. Like most of the MOAs, the maximum number of iterations ($Maxiter$) is considered as the stopping condition of the algorithm. If the current iteration number is smaller or equal to $Maxiter$, the algorithm will return to update the position of the observers. But if the current iteration number is larger than $Maxiter$, the DE-MEDT algorithm will terminate and report the best position of the observer found by the algorithm.

2.6 Pseudo-code of the DE-MEDT algorithm

The pseudo-code of the DE-MEDT algorithm is detailed in Algorithm 1.

Algorithm 1 Pseudo code of the DE-MEDT algorithm.

```
% Initialization phase
nOs: number of observers as candidate solutions
Maxiter: maximum number of iterations as stopping criterion of the algorithm
α and pa: algorithm-specific parameters of the algorithm
Initialize the observers' positions randomly using Eq. (2).
Evaluate the objective function value of observers' positions and sort them.
while (iter ≤ Maxiter) do
    Calculate the average position of the observers (M(x̄)) using Eqs (19) and (20).
    Determine the ED_i using Eq. (18).
    for every observer (i → nOs) do
        % Position updating of the observers based on DE equation
        Find the determinative agent (x_det)
        Calculate v_o, v_s, and v using Eqs. (11-13), respectively.
        Calculate the new position of the ith observer using Eqs. (9) and (10).
        Determine the scatter radius index of the current iteration (SRI^Iter) using Eq. (16)
        Normalize the SRI^Iter to the search space of the given problem using Eq. (17).
        Calculate the convergence index (CI) using Eq. (15).
        if r_1 < pa × (1 − CI) do
            % MEDT mechanism
```

Update the ith observer position using Eq. (14).
end if
Checking boundary conditions of design variables using Eqs. (21) and (22).
Evaluate the new position of the ith observer.
end for
Merge population of the current and previous iterations.
Sort merged population and select the first nOs observers of them.
$iter = iter + 1$;
end while
Report the observer with the best objective function value in the population.

3. DESIGN EXAMPLES

In this section, five large-scale truss structures are investigated to evaluate the performance of the DE-MEDT optimizer. The optimization results achieved by the DE-MEDT algorithm are compared with other optimization techniques existing in the literature.

Optimization aims to minimize the weight of these truss structures while satisfying some constraints on stress and displacement. Eq. (23) mathematically states a formulation for weight minimization of these structures under their certain design constraints.

$$\begin{aligned} &\text{Find } \{X\} = [x_1, x_2, \dots, x_{ng}]; \quad x_i \in D \\ &To\ minimize\ \ W(\{X\}) = \sum_{i=1}^{ng} x_i \sum_{j=1}^{nm(i)} \rho_j L_j \\ &subject\ to:\ g_k(\{X\}) \le 0; \quad k = 1,2, \dots, nc \end{aligned} \tag{23}$$

where $\{X\}$ is the vector of design variables; ng is the number of element groups (i.e., number of design variables); x_i is the value of the cross-sectional area in the ith element group; D denotes to the design space, and the values of cross-sectional areas of truss members are selected from it; $W(\{X\})$ is the weight of the truss structure; $nm(i)$ is the number of truss elements in the ith element group; ρ_j and L_j are the material density and length of the jth member of the ith element group, respectively; $g_k(\{X\})$ represents the kth design constraint, and nc is the number of design constraints existing in the truss optimization problem. It is worth mentioning that, in discrete design optimization of truss structures, the cross-sectional areas are selected from a discrete set of sections. However, in continuous design optimization, the cross-sectional areas can take a value from a continuous range.

This paper uses the penalty method to handle the design constraints. According to this approach, a penalty term is added to the objective function, as in Eq. (24).

$$W_P(\{X\}) = (1 + \varepsilon_1 . v)^{\varepsilon_2} \times W(\{X\}); \quad v = \sum_{i=1}^{nc} max[0, g_k(\{X\})] \tag{24}$$

where $W_P(\{X\})$ is the penalized weight of the truss structure, v denotes to the sum of the violation ratios of design variables. The constant ε_1 is set equal to 1, and ε_2 is obtained as follows:

$$\varepsilon_2 = 1.5 + 1.5 \times \frac{iter}{Maxiter} \tag{25}$$

The truss structures optimized by DE-MEDT in the following subsections are a 160-bar transmission tower, a 693-bar double-layer barrel vault, a 942-bar truss tower, a 1016-bar double-layer grid, and a 1410-bar dome-shaped structure. In all test examples, the population size (number of observers) is equal to $nOs = 30$. The algorithm-specific parameters (α and pa) are similar to the source paper, which are 10 and 0.5, respectively. Due to the stochastic nature of metaheuristics, 20 independent runs ($NIRs = 20$) are performed to get statistically meaningful results. The computer codes for the algorithm and structures are prepared in the MATLAB software environment, and the truss structures are analyzed using the direct stiffness method.

3.1 A 160-bar transmission tower

Discrete sizing optimization of the truss structure is a challenging issue in structural design. In the first design example, the DE-MEDT optimizer is evaluated for sizing optimization of a 160-bar transmission tower with 38 discrete design variables. The schematic of the structure is shown in Fig. 3. Detailed information on this design example can be found in Refs [3, 22].

Table 1 compares the optimization results of DE-MEDT with Regional Genetic Algorithm (RGA) [23], Rank-Based Ant System (RBAS) [24], Adaptive Elitist Differential Evolution (aeDE) [3], Electromagnetism-like Firefly Algorithm (EFA) [25], Self-Adaptive Multi-Population-based Jaya (SAMP-Jaya) [22], and Improved Shuffled based Jaya (IS-Jaya) [22]. It can be seen that the lightest weight is acquired by aeDE and IS-Jaya (1,336.634 kg), which is a little less than the structural weight found by DE-MEDT (1,336.704 kg). The proposed DE-MEDT algorithm obtained a lower weight than RGA (1,337.442 kg), RBAS (1,348.905 kg), and SAMP-Jaya (1,337.043 kg). DE-MEDT requires 16,830 structural analyses ($NSAs = 16{,}830$) to find the best weight equal to 1,336.704 kg. However, RBAS, aeDE, EFA, and SAMP-Jaya require 90,000, 23,925, 16,870, and 17,780 structural analyses, respectively. Comparing the average, worst, and standard deviation of the proposed DE-MEDT with other optimization methods in Table 1 reveals that the DE-MEDT algorithm is one of the most robust and reliable optimizers. The convergence histories of the best, worst, and average of runs recorded for the DE-MEDT algorithm are shown in Fig. 4. This figure shows that 600 iterations with a maximum number of function evaluations equal to 18000 are enough to find the optimum weight of the structure, and the algorithm converges to a specified value after almost 500 iterations. Fig. 5 illustrates that the

design constraints of the problem have not been violated at the best-optimized design found by the DE-MEDT.

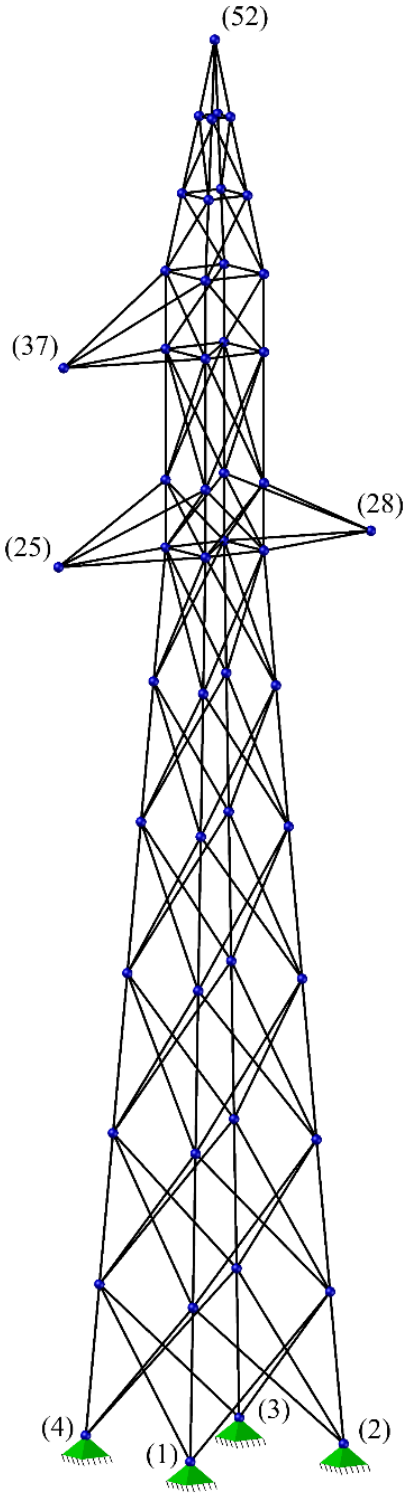


Figure 3. The 160-bar transmission tower

Table 1: Comparison of the results found by DE-MEDT and other considered metaheuristics for the 160-bar transmission tower

Element group	Optimal cross-sectional areas (cm^2)						
	Groenwold et al. [23]	Capriles et al. [24]	Ho-Huu et al. [3]	Le et al. [25]	Kaveh et al. [22]		Present work
	RGA	RBAS	aeDE	EFA	SAMP-Jaya	IS-Jaya	DE-MEDT
A1	19.03	19.03	19.03	19.03	19.03	19.03	19.03
A2	5.27	5.27	5.27	5.27	5.27	5.27	5.27
A3	19.03	19.03	19.03	19.03	19.03	19.03	19.03
A4	5.27	5.27	5.27	5.27	5.27	5.27	5.27
A5	19.03	19.03	19.03	19.03	19.03	19.03	19.03
A6	5.75	5.75	5.75	5.75	5.75	5.75	5.75
A7	15.39	15.39	15.39	15.39	15.39	15.39	15.39
A8	5.75	5.75	5.75	5.75	5.75	5.75	5.75
A9	13.79	13.79	13.79	13.79	13.79	13.79	13.79
A10	5.75	5.75	5.75	5.75	5.75	5.75	5.75
A11	5.75	5.75	5.75	5.75	5.75	5.75	5.75
A12	13.79	12.21	12.21	12.21	12.21	12.21	12.21
A13	6.25	6.25	6.25	6.25	6.25	6.25	6.25
A14	5.75	5.75	5.75	5.75	5.75	5.75	5.75
A15	2.66	3.47	3.88	3.88	3.47	3.88	3.88
A16	7.44	7.44	7.44	7.44	7.44	7.44	7.44
A17	1.84	1.84	1.84	1.84	1.84	1.84	1.84
A18	8.66	9.4	8.66	8.66	8.66	8.66	8.66
A19	2.66	2.66	2.66	2.66	2.66	2.66	2.66
A20	3.07	3.47	3.07	3.07	3.07	3.07	3.07
A21	2.66	3.07	2.66	2.66	3.47	2.66	2.66
A22	8.06	8.06	8.06	8.06	8.06	8.06	8.06
A23	5.27	5.27	5.75	5.75	5.75	5.75	5.75
A24	6.25	6.25	6.25	6.25	6.25	6.25	6.25
A25	5.75	5.75	5.75	6.25	5.75	5.75	6.25
A26	1.84	2.26	2.26	1.84	2.26	2.26	1.84
A27	4.79	4.79	4.79	4.79	4.79	4.79	4.79
A28	2.66	3.07	2.66	2.66	2.66	2.66	2.66
A29	3.47	3.47	3.47	3.47	3.47	3.47	3.47
A30	1.84	1.84	1.84	1.84	1.84	1.84	1.84
A31	2.26	3.88	2.26	2.26	2.26	2.26	2.26
A32	3.88	3.88	3.88	3.88	3.88	3.88	3.88
A33	1.84	1.84	1.84	1.84	1.84	1.84	1.84
A34	1.84	2.26	1.84	1.84	1.84	1.84	1.84
A35	3.88	3.88	3.88	3.88	3.88	3.88	3.88
A36	1.84	2.26	1.84	1.84	1.84	1.84	1.84
A37	1.84	3.47	1.84	1.84	1.84	1.84	1.84
A38	3.88	3.88	3.88	3.88	3.88	3.88	3.88
Best weight (kg)	1,337.442	1,348.905	1,336.634	1,336.704	1,337.043	1,336.634	1,336.704
Average weight (kg)	N/A	1,367.5275	1,355.875	1,372.551	1,355.328	1,342.807	1,340.585
Worst weight (kg)	N/A	1,401.6323	1,410.611	1,429.253	1,420.340	1,366.933	1,345.188
Standard deviation (kg)	N/A	N/A	18.805	34.706	20.691	8.649	2.265
NSAs	N/A	90,000	23,925	16,870	17,780	11,740	16,830

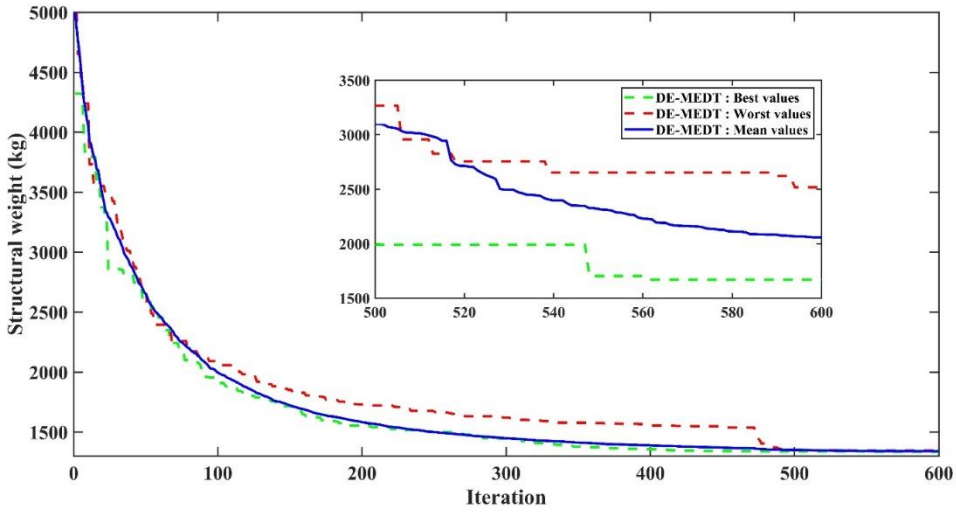


Fig. 4 Convergence curve of DE-MEDT for the 160-bar transmission tower

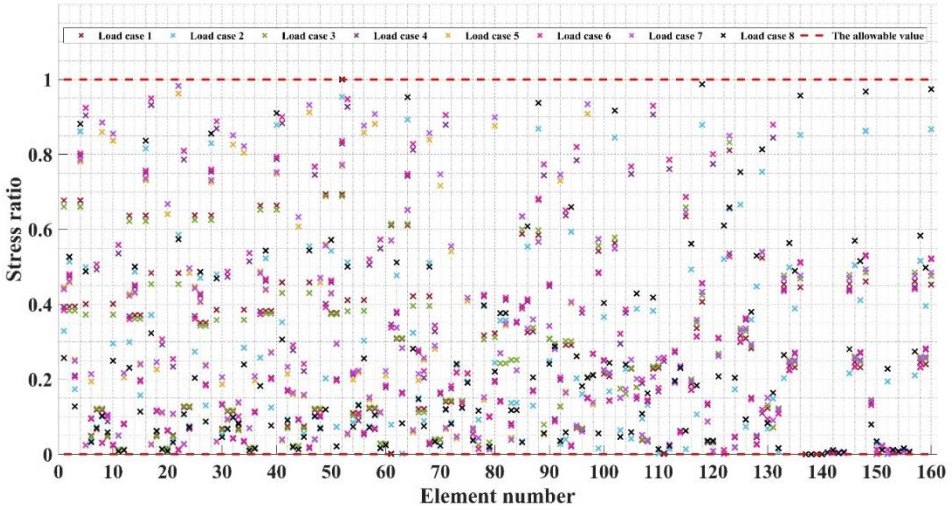


Fig. 5 Stress ratio values evaluated at the best-optimized design found by the DE-MEDT for the 160-bar transmission tower

3.2 A 693-bar double-layer barrel vault

The second design example examined in this research item is the weight minimization of a 693-bar double layer barrel vault, shown in Fig. 6. Due to the structural symmetry, all 693 members of the structure are grouped into 23 element groups. Design variables are the cross-sectional areas of the structure elements selected from 37 steel pipe sections from AISC-LRFD [26]. For further details, one can refer to Refs [27, 28].

Table 2 presents the optimization results obtained by DE-MEDT, Bat-Inspired (BI) [29], Modified Big Bang–Big Crunch (BB-BC) [30], Magnetic Charged System Search (MCSS) and its improved variant (IMCSS) [31], Enhanced Colliding Bodies Optimization (ECBO)

[32], and Multi Design Variable Configurations-Upper bound Vibrating Particles System (MDVC-UVPS) [32]. In this table, ST, EST, and DEST abbreviations stand for standard weight, extra strong, and double-extra strong, respectively. From the table, it can be observed that MDVC-UVPS achieved a lower weight than other optimization methods. The design optimized by DE-MEDT gives the weight of 9,107.4 lb, which is only 0.18 % heavier than the obtained weight by MDVC-UVPS. Comparing the average weight and standard deviation found by the DE-MEDT with BI, ECBO, and MDVC-UVPS algorithms indicate that the DE-MEDT algorithm is the most reliable optimization technique. Fig. 7 depicts the convergence histories of the best, worst, and average of runs recorded by the DE-MEDT algorithm. This figure shows that performing more than 600 iterations does not improve structural weight significantly, and after almost 300 iterations, the algorithm starts to converge to a specified value. Fig. 8 shows that all design constraints of the problem have been satisfied at the best-optimized design obtained by the DE-MEDT.

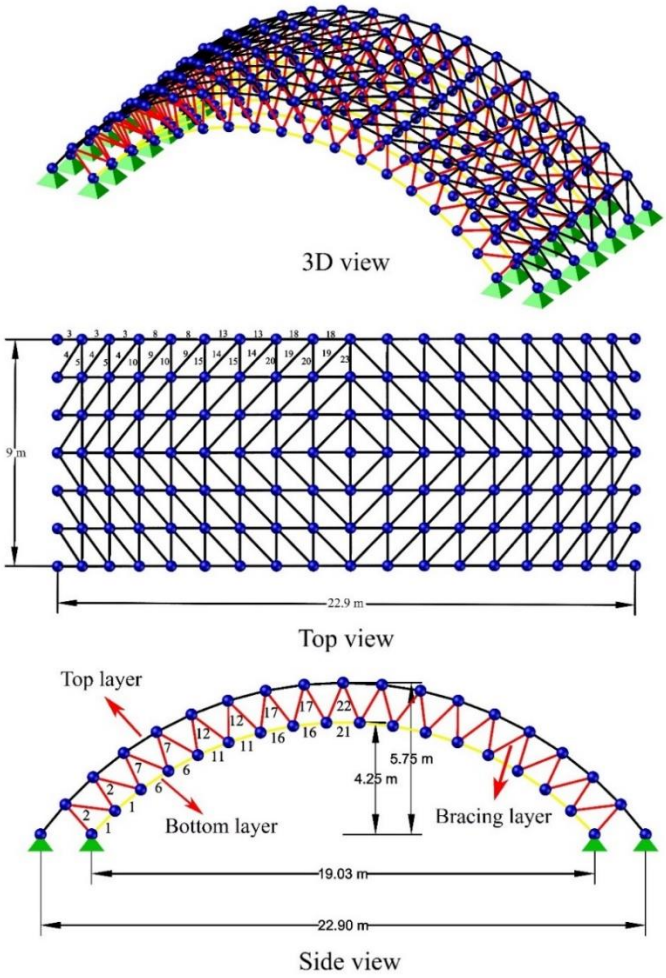


Figure 6. The 693-bar double-layer barrel vault

Table 2: Comparison of the results found by DE-MEDT and other considered metaheuristics for the 693-bar double-layer barrel vault

Element group	Hasançebi et al. [29]	Hasançebi and Kazemzadeh Azad [30]	Kaveh et al. [31]		Kaveh and Ilchi Ghazaan [32]		Present work
	BI	MBB-BC	MCSS	IMCSS	ECBO	MDVC-UVPS	DE-MEDT
1	EST 3 1/2	EST 3 1/2	EST 3	EST 3 1/2	ST 4	ST 4	ST 4
2	ST 1	ST 1	ST 1	ST 1	ST 1	ST 1	ST 1
3	ST 3/4	ST 3/4	EST 3/4	EST 1	ST 3/4	ST 3/4	ST 3/4
4	ST 1	ST 1	EST 1/2	ST 3/4	ST 1	ST 1	ST 1
5	ST 3/4	ST 3/4	EST 1/2	ST 1	ST 3/4	ST 3/4	ST 3/4
6	EST 3 1/2	EST 3 1/2	EST 3	DEST 2	ST 3	ST 3 1/2	ST 3 1/2
7	ST 1	ST 1	EST 1 1/4	ST 1	ST 1	ST 1	ST 1
8	ST 1	ST 1	ST 1	ST 1 1/4	ST 1	ST 1	EST 3/4
9	ST 1	ST 1	ST 3/4	EST 1/2	ST 1	ST 1	ST 1
10	ST 3/4	ST 3/4	EST 1/2	ST 1/2	ST 3/4	ST 3/4	ST 3/4
11	DEST 2	ST 3	EST 2 1/2	ST 3	EST 2	EST 2 1/2	ST 3
12	ST 1 1/2	ST 1 1/2	EST 1 1/2	EST 1 1/4	ST 1 1/4	ST 1	ST 1
13	ST 2	EST 1 1/2	ST 2 1/2	EST 2	EST 2	ST 1 1/2	ST 1 1/4
14	ST 1	ST 1	ST 3/4	ST 1/2	ST 1	ST 1	ST 1
15	ST 3/4	ST 3/4	ST 3/4	ST 3/4	ST 3/4	ST 3/4	ST 3/4
16	EST 1 1/2	EST 1 1/4	ST 1 1/4	EST 1 1/4	ST 1	EST 1 1/4	ST 1 1/2
17	EST 1	ST 1 1/4	ST 1 1/2	ST 1 1/2	ST 1	ST 1	ST 1 1/4
18	ST 2 1/2	ST 3	ST 3	ST 3	ST 3	EST 2	EST 2
19	ST 1	ST 1	EST 3/4	ST 3/4	ST 1	ST 1	ST 1
20	ST 3/4	ST 3/4	ST 1/2	ST 3/4	ST 3/4	ST 3/4	ST 3/4
21	ST 1	ST 1	ST 1 1/4	ST 1	ST 3/4	ST 1	ST 1
22	ST 3/4	ST 3/4	EST 3/4	EST 1	ST 3/4	ST 1	ST 3/4
23	ST 3/4	ST 3/4	ST 3/4	EST 3/4	ST 3/4	ST 3/4	ST 3/4
Best weight (lb)	10,564.84	10,595.33	10,812.39	10,550,86	9,240.5	9,091.1	9,107.4
Average weight (lb)	10,595.66	N/A	N/A	N/A	9,577	9,475	9,288.7
Worst weight (lb)	N/A	N/A	N/A	N/A	N/A	N/A	9,421.5
Standard deviation (lb)	11.12	N/A	N/A	N/A	505	765	96.4
NSAs	36,300	50,000	14,300	9,200	16,720	4,120	17,880

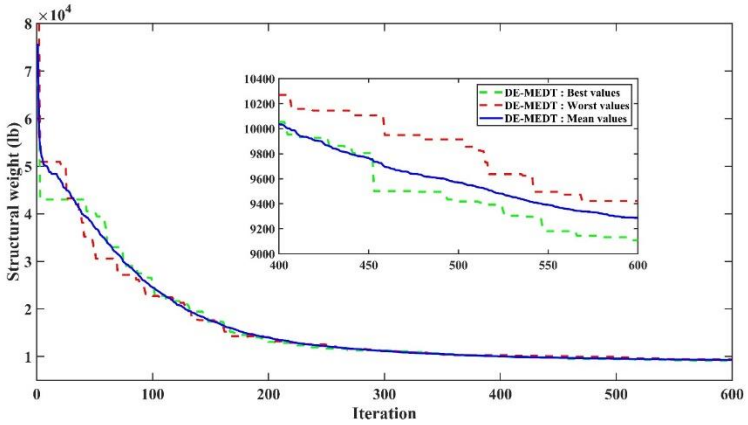
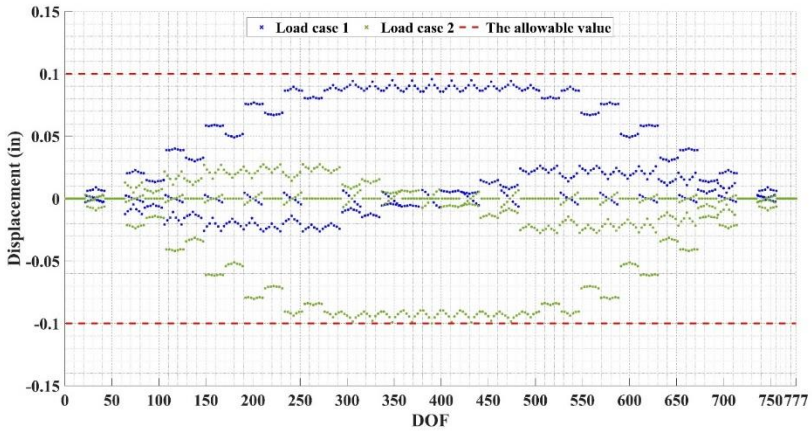
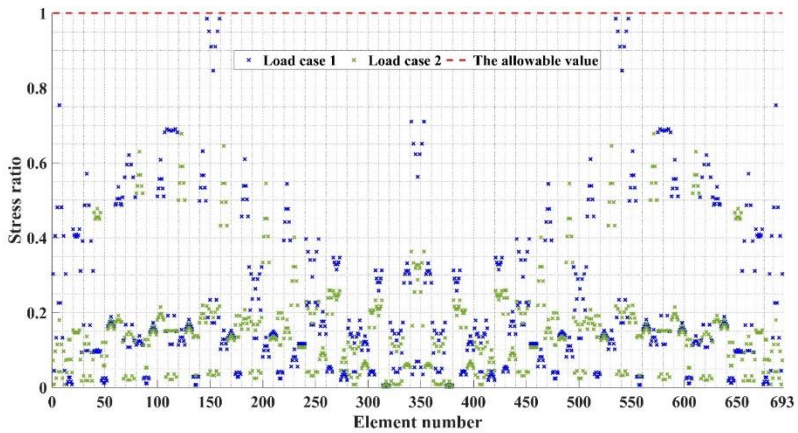


Figure 7. Convergence curve of DE-MEDT for the 693-bar double-layer barrel vault



(a)



(b)

Figure 8. Constraint boundaries of the 693-bar double-layer barrel vault evaluated at the best-optimized design by the DE-MEDT: (a) Displacement, (b) Stress ratio

3.3. A 942-bar truss tower

In the third design example, the 942-bar truss tower structure displayed in Fig. 9 is considered to evaluate the performance of the present algorithm when a large number of design variables participate in the optimization process. This tower structure consists of 924 structural members and 244 nodes. According to the schematic of the structure given in Fig. 9, all 942 elements are grouped into 59 element groups. Optimization variables are the cross-sectional area of the elements selected from a discrete set. Further details on this optimization problem can be found in Ref [22].

Table 3 summarizes the comparison results obtained by the present DE-MEDT and other optimization methods, which are Chaotic Firefly Algorithm based on Gaussian map (CGFA) [33], SAMP-Jaya [22], and IS-Jaya [22]. From this table, it can be concluded that DE-MEDT gives the best optimum weight (138,437 lb). A careful examination of this table also reveals that the statistical results of the present DE-MEDT metaheuristic are better than other considered optimizers. Convergence histories of the best, worst, and average of runs recorded by the DE-MEDT algorithm are shown in Fig. 10. This figure shows that performing 2000 iterations is enough for completing the optimization process, and the present algorithm converges to the specified value after almost 1200 iterations. Displacement and stress values evaluated at the best-optimized design are given in Fig. 11. From this figure, it can be concluded that the design constraints of the problem at the best-optimized design have not been violated.

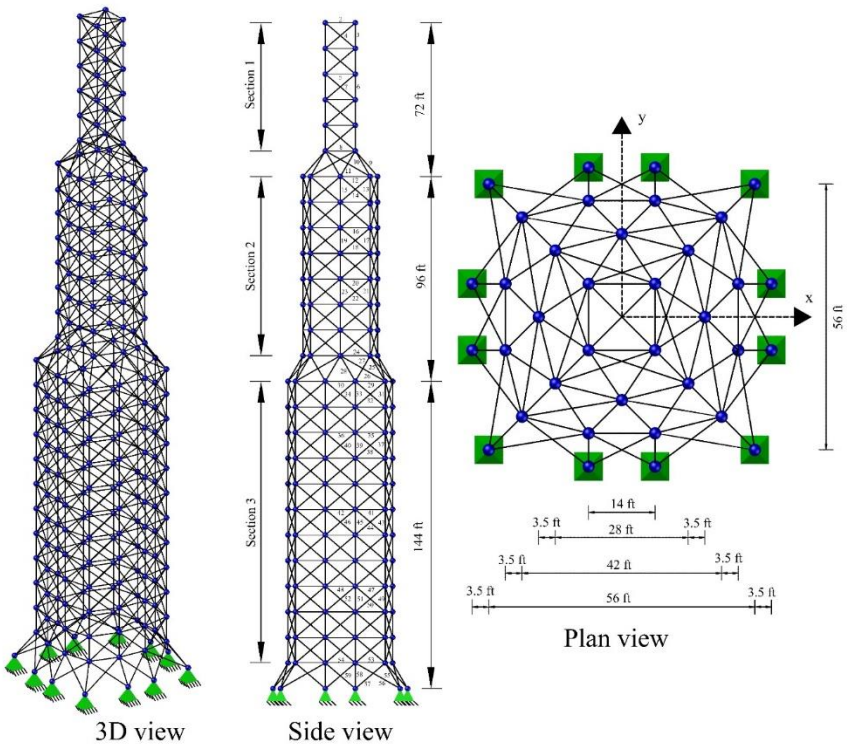


Figure 9. The 942-bar truss tower

Table 3: Comparison of the results found by DE-MEDT and other considered metaheuristics for the 942-bar truss tower

Element group	Optimal cross-sectional areas (in.^{2})			
	Kaveh et al. [33]	Kaveh et al. [22]		Present work
	CGFA	SAMP-Jaya	IS-Jaya	DE-MEDT
A1	1	1	1	1
A2	1	1	1	1
A3	1	3	4	4
A4	1	1	2	2
A5	1	1	1	1
A6	14	15	15	16
A7	4	3	3	3
A8	5	4	6	7
A9	5	7	6	5
A10	22	17	28	20
A11	1	3	5	3
A12	4	7	7	6
A13	19	15	16	16
A14	2	2	2	2
A15	4	5	5	4
A16	1	2	1	1
A17	21	22	22	23
A18	3	3	3	3
A19	14	9	9	8
A20	1	1	1	1
A21	35	29	29	27
A22	3	4	5	3
A23	18	16	18	18
A24	24	23	26	25
A25	36	39	40	42
A26	1	12	3	1
A27	11	5	13	13
A28	14	11	15	17
A29	14	14	16	15
A30	23	16	17	14
A31	38	37	38	41
A32	3	3	3	3
A33	2	3	4	3
A34	3	3	3	3
A35	1	1	1	1
A36	1	1	1	1
A37	70	59	62	60
A38	3	4	3	4
A39	2	2	2	2
A40	3	3	3	3
A41	1	1	1	1
A42	1	1	8	1
A43	91	83	69	74
A44	3	4	5	3
A45	2	2	1	2

A46	2	3	5	3
A47	1	1	1	1
A48	1	1	1	1
A49	102	96	76	96
A50	4	3	3	4
A51	1	2	7	1
A52	3	4	5	4
A53	10	9	18	8
A54	11	5	16	9
A55	46	44	57	44
A56	1	1	1	1
A57	65	62	51	66
A58	3	3	7	4
A59	1	3	4	1
Best weight (lb)	141,860	139,744	138,689	138,437
Average weight (lb)	144,231	170,279	142,903	140,746
Worst weight (lb)	147,325	236,898	150,722	143,177
Standard deviation (lb)	3,342	28,367	3,171	1,484
NSAs	32,500	56,920	53,420	59,820

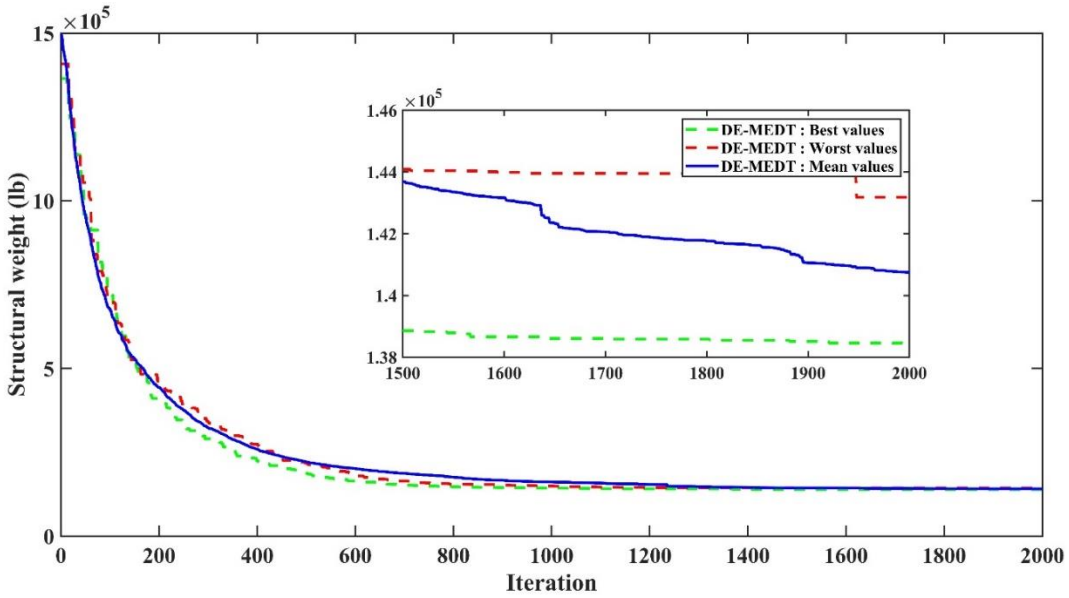


Figure 10. Convergence curve of DE-MEDT for the 942-bar truss tower

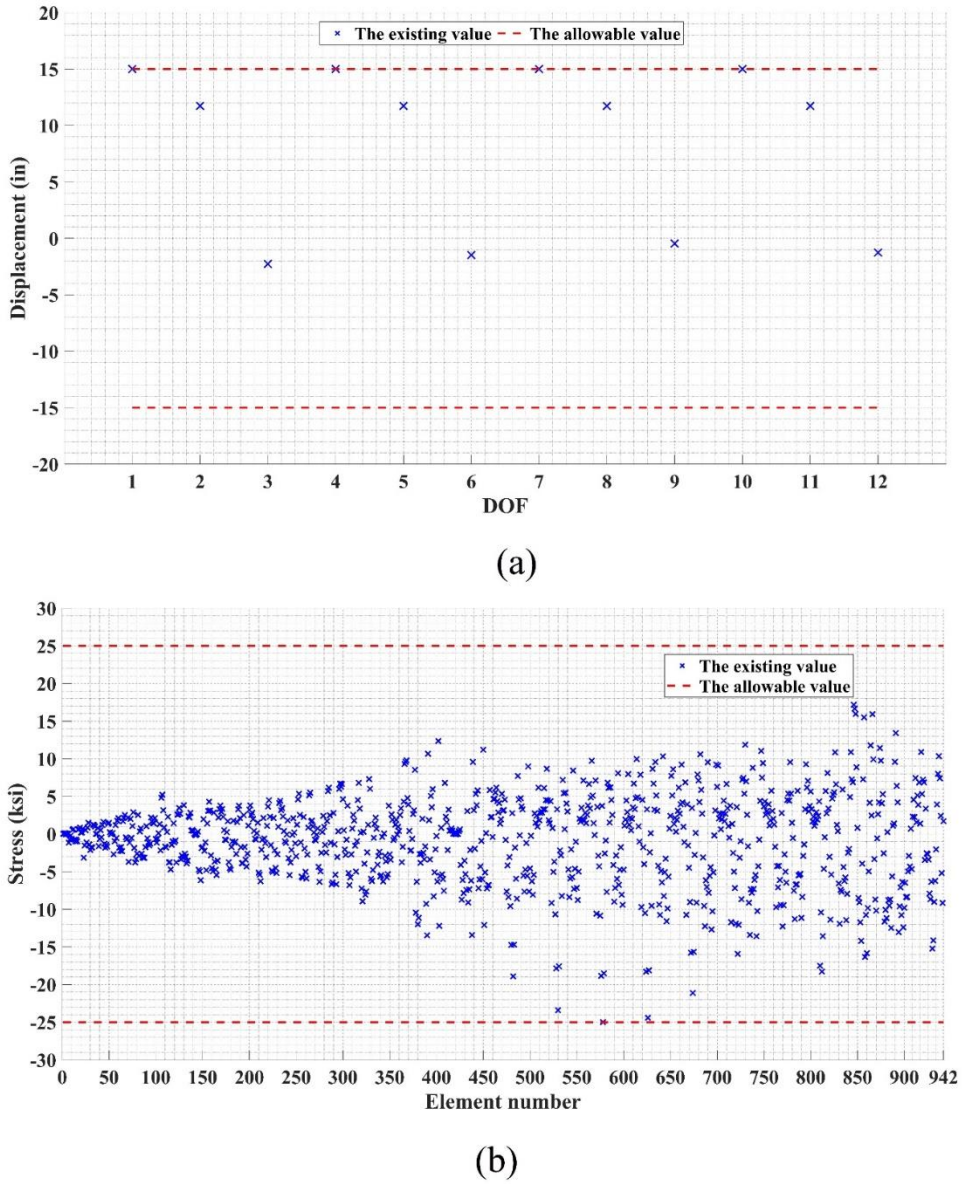


Figure 11. Constraint boundaries of the 942-bar truss tower evaluated at best-optimized design by the DE-MEDT: (a) Displacement, (b) Stress.

3.4 A 1016-bar double-layer grid

A 1016-bar double layer grid is taken as the fourth test example to evaluate the performance of the proposed DE-MEDT optimizer. The structure consists of 1016 bar elements and 320 nodes. Fig. 12 displays a schematic view of the structure, where the bottom, top, and web members are grouped into 11, 9, and 5 element groups, respectively, due to structural symmetry. The distance between the top and bottom layers is equal to 3 m, and the structure

has a span length of 40×40 m. The top layer nodes of the grid are subjected to a concentrated vertical load of 30 KN. The cross-section areas of the members are selected from the list of 37 steel pipe sections from AIS-LRFD [26]. Design constraints of the strength and slenderness of the members are based on the requirements of AISC-LRFD [26], and the limitation for displacement must not be larger than (span length)/600 for all nodes in the vertical direction. Further details on this optimization problem can be found in ref [34].

Table 4 presents the comparisons between the results of DE-MEDT and those of other optimization methods, including ECBO [32], MDVC-UVPS [32], PRSSOA [35], and ESSOA [34]. Based on the results given in this table, it can be observed that DE-MEDT is capable of finding the best weight (65,125 kg) in comparison to other considered metaheuristics. Moreover, the statistical results gained by the DE-MEDT optimizer reveal that the present algorithm is the most reliable method compared to other optimization methods. The convergence histories of the best, worst, and average of runs are plotted in Fig 13. A close examination of this figure shows that performing 600 iterations is completely enough, and the algorithm starts to converge a specific amount after approximately 300 iterations. Constraint boundaries of the structure are assessed at the best optimum design and shown in Fig. 14. From this figure, it can be observed that all design constraints of the problem have been satisfied.

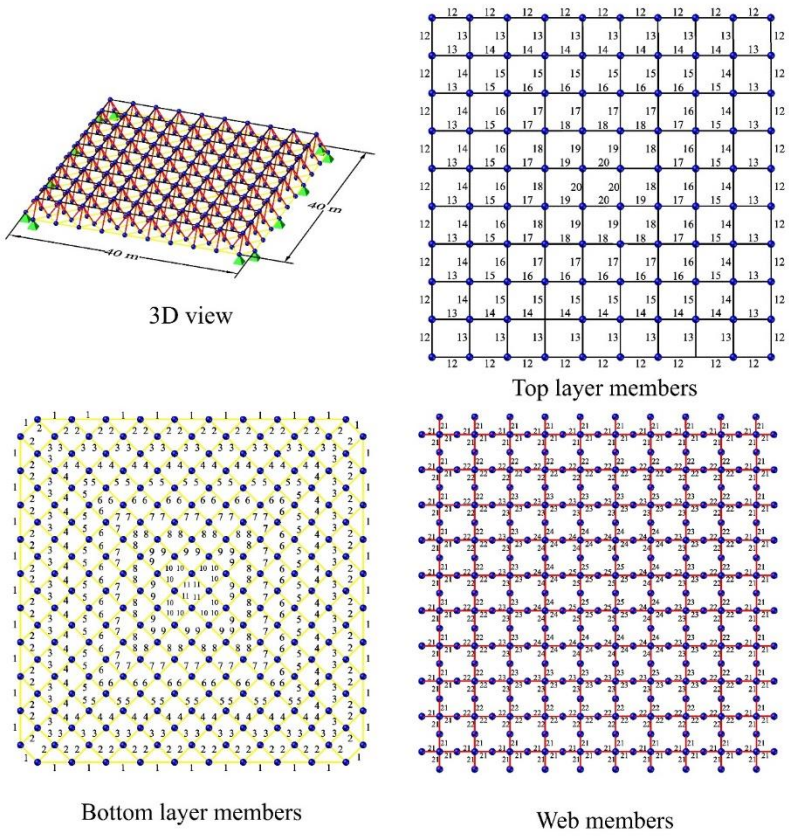


Figure 12. The 1016-bar double-layer grid

Table 4: Comparison of the results found by DE-MEDT and other considered metaheuristics for the 1016-bar double-layer grid.

	Kaveh and Ilchi Ghazaan [32]		Kaveh et al. [35]	Kaveh et al. [34]	Present work
Element group	ECBO	MDVC-UVPS	PRSSOA	ESSOA	DE-MEDT
1	EST 5	DEST 4	EST 5	ST 6	ST 2 1/2
2	EST 5	DEST 3	EST 4	ST 5	DEST 2
3	ST 3	ST 3 1/2	EST 3	EST 3	DEST 2 1/2
4	ST 3 1/2	ST 2 1/2	ST 2 1/2	EST 2 1/2	DEST 3
5	ST 2 1/2	ST 3	ST 3	ST 3	ST 6
6	ST 2	EST 1 1/2	EST 1 1/2	EST 1 1/2	DEST 5
7	DEST 2	EST 1 1/2	EST 1	EST 1 1/2	ST 8
8	DEST 2	EST 2 1/2	ST 2 1/2	ST 2 1/2	ST 6
9	EST 2	ST 3 1/2	EST 2	EST 3	EST 4
10	ST 6	DEST 2	ST 3 1/2	EST 2 1/2	DEST 2 1/2
11	ST 2	DEST 2 1/2	ST 4	EST 4	ST 5
12	EST 8	EST 8	ST 10	ST 10	ST 1
13	EST 3 1/2	EST 4	ST 6	ST 4	ST 4
14	ST 5	ST 4	ST 5	ST 5	ST 2 1/2
15	ST 4	ST 5	ST 5	EST 4	EST 2 1/2
16	EST 5	ST 4	ST 5	ST 6	ST 2 1/2
17	ST 5	ST 6	ST 6	EST 4	ST 2 1/2
18	EST 5	ST 6	EST 5	ST 5	DEST 2
19	EST 5	EST 6	EST 5	EST 6	EST 1 1/4
20	ST 8	EST 6	DEST 4	EST 6	ST 2
21	ST 5	ST 5	ST 6	ST 6	ST 2 1/2
22	ST 3	ST 3 1/2	ST 3 1/2	ST 3 1/2	DEST 2
23	EST 2 1/2	EST 2 1/2	ST 3 1/2	ST 3 1/2	DEST 2 1/2
24	ST 5	ST 2 1/2	ST 2 1/2	EST 2 1/2	DEST 3
25	ST 4	ST 2 1/2	EST 1 1/2	EST 1 1/2	ST 6
Best weight (kg)	67,839	65,826	67,407	67,079	65,125
Average weight (kg)	73,042	70,488	70,054	70,408	68,403
Worst weight (kg)	N/A	N/A	N/A	80,828	71,565
Standard deviation (kg)	9,158	5,018	1,864	2,703	1,663
NSAs	15,760	4,234	12,020	11,680	16,290

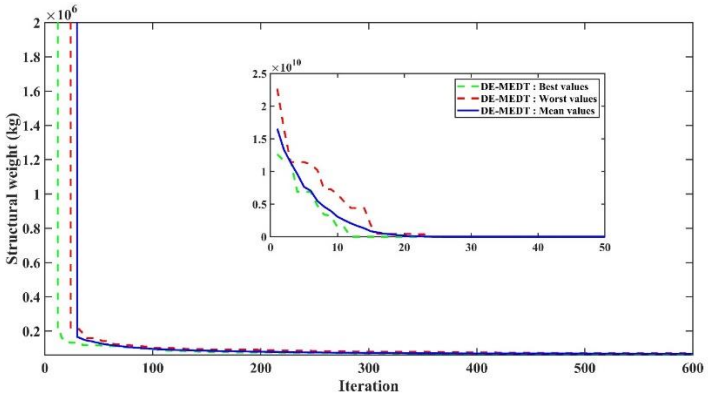


Figure 13. Convergence curve of DE-MEDT for the 1016-bar double-layer grid

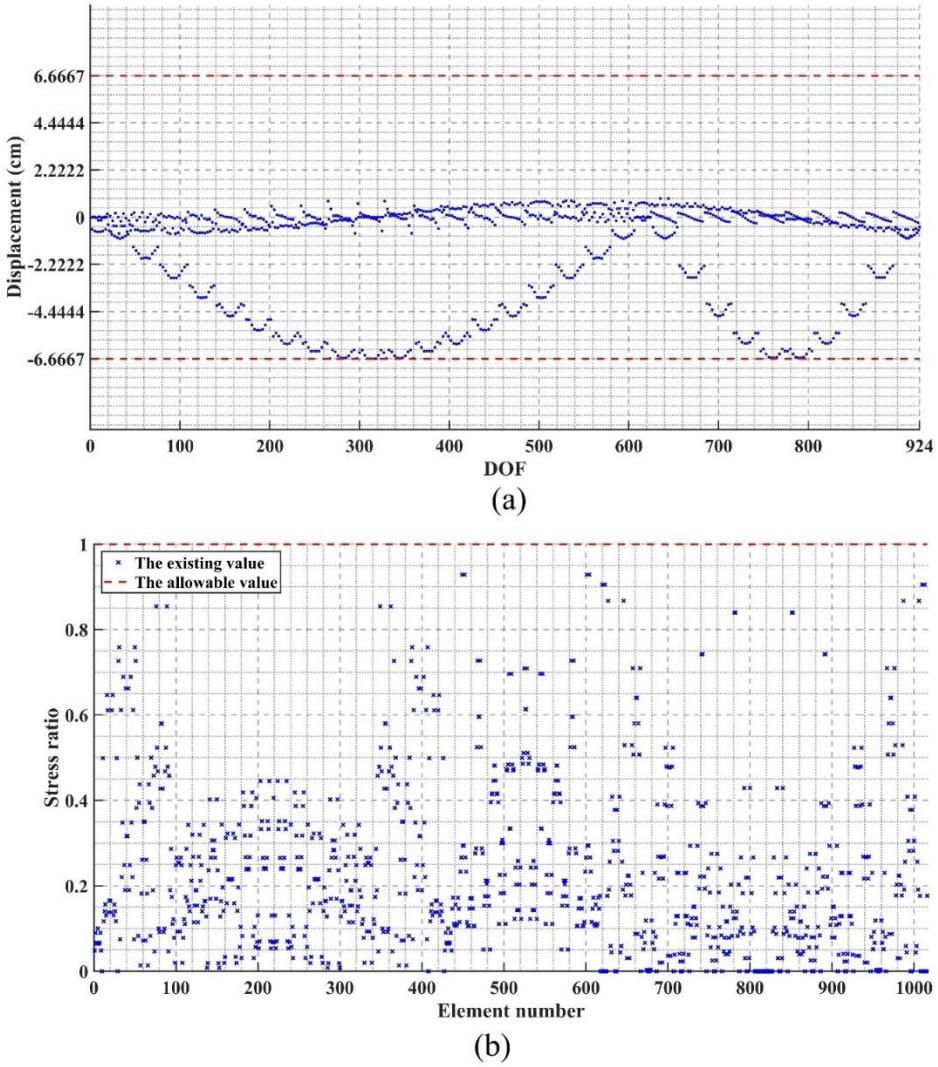


Figure 14. Constraint boundaries of the 1016-bar double-layer grid evaluated at the best-optimized design by the DE-MEDT: (a) Displacement, (b) Stress ratio

3.5 A 1410-bar dome-shaped structure

The 140-bar dome-shaped structure optimized as the last design example is shown in Fig. 15. The dome-shaped structure is composed of 1410 bar elements and 390 nodes. A substructure of the dome with further detail for nodal numbering is also displayed in Fig. 15. Due to the concept of symmetry existing in the configuration processing of the structure, 1410 dome elements are grouped into 47 element groups. Thus, the problem has 47 sizing variables. These variables are the cross-sectional areas of the dome elements selected from a continuous range. The range's minimum and maximum permissible values are equal to 1×10^{-4} m^2 and 100×10^{-4} m^2, respectively. For further details, one can refer to Ref [34].

Table 5 compares the optimized design obtained by DE-MEDT with those achieved by

other optimization methods presented by Kaveh and Ilchi Ghazaan [32] (including ECBO and MDVC-UVPS) and Kaveh et al. [34], which is ESSOA. A close examination of this table shows that the present algorithm obtained the lightest design, which is 7,268.2 kg. Furthermore, comparing the statistical results of the DE-MEDT algorithm with those gained by other optimization methods shows that the present algorithm is more robust and reliable than other optimization techniques. Convergence curves of the best, worst, and average of runs reported for the DE-MEDT algorithm are shown in Fig. 16. From this figure, it can be seen that 1000 iterations are almost enough to complete the optimization process. Fig. 17 displays constraint boundaries of the dome-shaped structure evaluated at the best optimum design. It can be observed that the problem's constraints have not been violated.

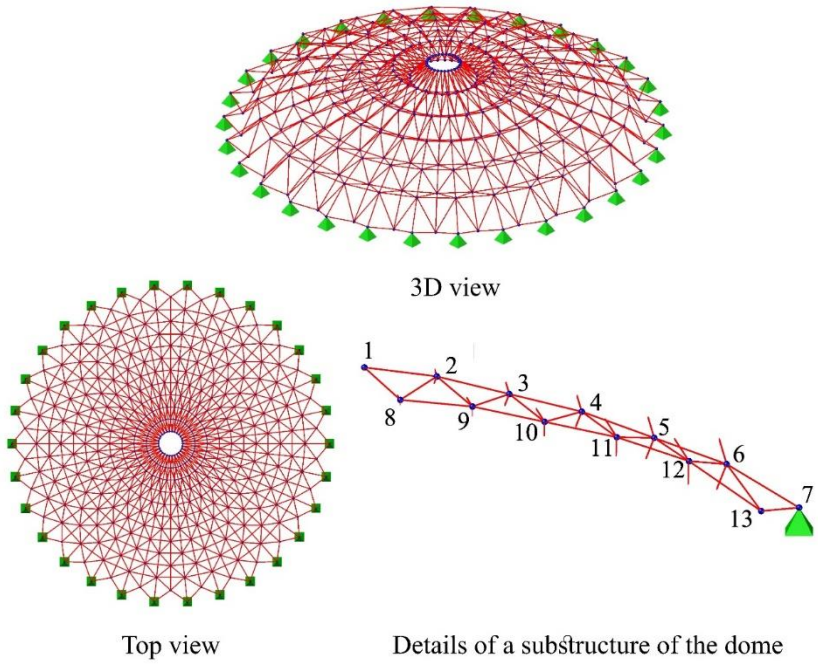


Figure 15. The 1410-bar dome-shaped structure

Table 5: Comparison of the results found by DE-MEDT and other considered metaheuristics for the 1410-bar dome-shaped structure

Element number (nodes)	Kaveh and Ilchi Ghazaan [32]		Kaveh et al. [34]	Present work
	ECBO	MDVC-UVPS	ESSOA	DE-MEDT
1 (1–2)	5.217	4.8489	4.8298	4.7884
2 (1–8)	2.213	1.5104	2.0165	1.7750
3 (1–14)	4.0413	4.3939	7.5790	1.0327
4 (2–3)	5.3523	4.8489	4.8331	4.7663
5 (2–8)	2.8635	2.3413	3.8613	1.5713
6 (2–9)	1.8832	1.6246	1.8555	2.0758
7 (2–15)	1.0007	4.3939	1.5531	2.8807
8 (3–4)	6.4681	4.8489	5.3270	5.6240

9 (3–9)	1.2068	2.1707	2.8348	3.0420
10 (3–10)	1.738	1.6765	2.3982	2.5317
11 (3–16)	12.5144	4.3939	2.8265	4.6835
12 (4–5)	6.3101	7.6688	6.1447	5.6677
13 (4–10)	1.7218	2.4287	1.4803	2.8792
14 (4–11)	2.4362	1.8282	2.3323	2.1620
15 (4–17)	3.5615	5.5832	4.0043	3.1916
16 (5–6)	6.1832	7.6688	6.5057	6.6086
17 (5–11)	2.7977	2.5749	3.0395	2.5833
18 (5–12)	4.1412	3.6629	4.0052	3.8903
19 (5–18)	4.1542	5.5832	4.2089	4.7851
20 (6–7)	7.9148	7.6688	7.8692	7.9037
21 (6–12)	5.894	3.7234	3.0147	3.2501
22 (6–13)	3.3083	3.1638	3.5449	3.2243
23 (6–19)	6.6223	5.5832	5.7876	5.7041
24 (7–13)	3.6804	3.64	4.1487	4.8753
25 (8–9)	4.8207	6.1741	5.0264	4.7990
26 (8–14)	1.5864	1.5104	2.7103	1.9261
27 (8–15)	2.5913	2.3413	3.5774	1.8953
28 (8–21)	1.0843	4.0242	1.1653	1.0601
29 (9–10)	5.9325	6.1741	5.1417	4.8014
30 (9–15)	3.0351	1.6246	1.8443	1.8212
31 (9–16)	1.2356	2.1707	2.4992	2.6295
32 (9–22)	1.708	4.0242	2.3271	6.2965
33 (10–11)	4.8743	6.3156	5.2449	4.7525
34 (10–16)	3.429	1.6765	2.2090	2.8573
35 (10–17)	1.9623	2.4287	1.7165	2.5041
36 (10–23)	2.7079	4.8511	3.9555	3.8466
37 (11–12)	5.0557	6.3156	5.3902	5.4336
38 (11–17)	4.1289	1.8282	2.4328	1.9125
39 (11–18)	3.4292	2.5749	3.0903	3.0692
40 (11–24)	4.9348	4.8511	3.8433	4.2145
41 (12–13)	7.3564	6.3156	6.4713	6.5255
42 (12–18)	4.4329	3.6629	3.8202	3.7177
43 (12–19)	3.3212	3.7234	3.3478	3.1441
44 (12–25)	4.9391	4.8511	5.2009	4.9215
45 (13–19)	3.7342	3.1638	3.4564	3.2049
46 (13–20)	4.1154	3.64	3.9634	3.8039
47 (13–26)	5.0799	4.8511	3.8686	3.6826
Best weight (kg)	7,860.01	7,661.64	7,331.6	7,268.2
Average weight (kg)	8,250.20	8,106.52	7,602.3	7,835.7
Worst weight (kg)	N/A	N/A	8,049.4	8,488.1
Standard deviation (kg)	409.09	244.08	185.5	363.5
NSAs	19,840	16,308	19,400	29,910

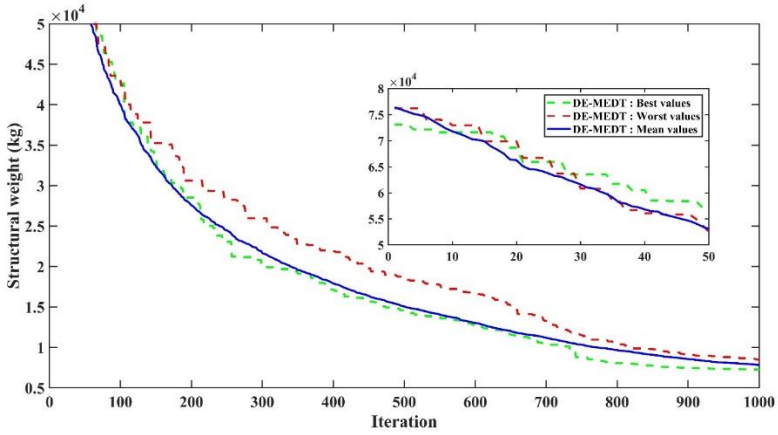
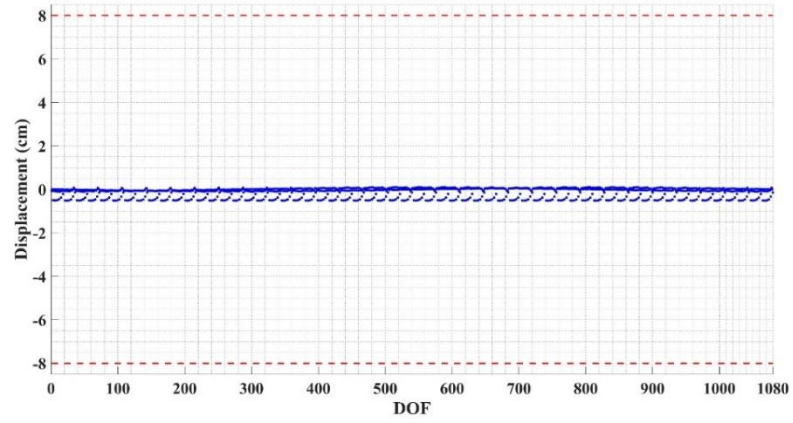
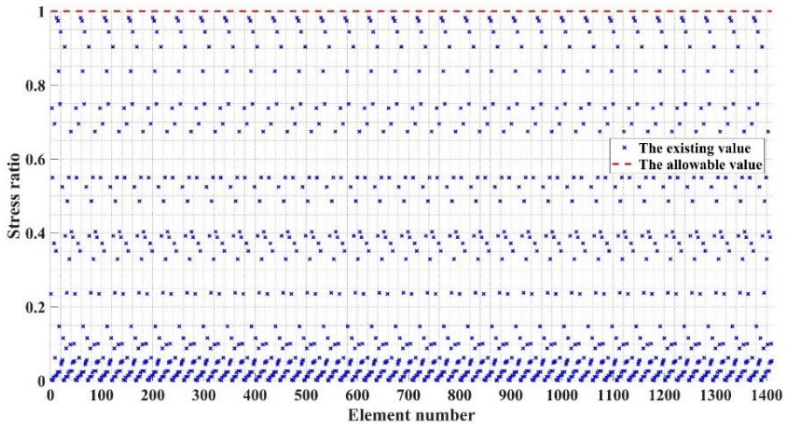


Figure 16. Convergence curve of DE-MEDT for the 1410-bar dome-shaped structure



(a)



(b)

Fig. 17 Constraint boundaries of the 1410-bar dome-shaped structure evaluated at the best-optimized design by the DE-MEDT: (a) Displacement, (b) Stress ratio

4. CONCLUDING REMARKS

In this paper, a new physically inspired metaheuristic called DE-MEDT has been successfully applied to solve large-scale truss optimization problems with discrete and continuous design variables. The Doppler Effect-Mean Euclidian Distance Threshold (DE-MEDT) algorithm is a recently proposed metaheuristic developed based on the Doppler Effect (DE) phenomenon and Mean Euclidian Distance Threshold (MEDT) mechanism. Equipping the DE-MEDT algorithm with the MEDT mechanism improves the quality of the candidate solutions and decreases the possibility of trapping into local optima. Five large-scale truss structures have been investigated to verify the efficiency and effectiveness of the DE-MEDT algorithm. The structures include a 160-bar transmission tower, a 693-bar double-layer barrel vault, a 942-bar truss tower, a 1016-bar double-layer grid, and a 1410-bar dome. The first four investigated structures have discrete design variables, and the cross-sectional areas of truss members are selected from a discrete set. However, the design variables of the last structure can take a value from a continuous range, and the cross-sectional areas of its elements are selected from a continuous interval. Design optimization aims to minimize the weight of these structures while satisfying design requirements. The optimization results found by the DE-MEDT have been compared with those of other metaheuristic algorithms existing in the literature. Comparison of the results found by the DE-MEDT algorithm with other existing metaheuristics proposed in the literature indicated that the DE-MEDT metaheuristic is comparable to or better than many other optimization techniques.

REFERENCES

1. Kaveh A. *Advances in Metaheuristic Algorithms for Optimal Design of Structures*, Springer, 3rd edition, Cham, Switzerland, 2021.
2. Dede T, Bekiroğlu S, Ayvaz Y. Weight minimization of trusses with genetic algorithm, *Appl Soft Comput* 2011; **11**(2): 2565-75.
3. Ho-Huu V, Nguyen-Thoi T, Vo-Duy T, Nguyen-Trang T. An adaptive elitist differential evolution for optimization of truss structures with discrete design variables, *Comput Struct* 2016; **165**: 59-75.
4. Mortazavi A, Toğan V, Nuhoğlu A. Weight minimization of truss structures with sizing and layout variables using integrated particle swarm optimizer, *J Civil Eng Manag* 2017; **23**(8): 985-1001.
5. Kaveh A, Javadi SM. Shape and size optimization of trusses with multiple frequency constraints using harmony search and ray optimizer for enhancing the particle swarm optimization algorithm, *Acta Mech* 2014, **225**(6), 1595-605.
6. Kaveh A, Zakian P. Enhanced bat algorithm for optimal design of skeletal structures, *Asian J Civil Eng (Build Hous)* 2014, **15**(2), 179-212.
7. Kaveh A, Zakian P. Improved GWO algorithm for optimal design of truss structures, *Eng Comput* 2018; **34**(4): 685-707.

8. Awad R. Sizing optimization of truss structures using the political optimizer (PO) algorithm, *Struct* 2021; **33**: 4871-94.
9. Salcedo-Sanz S. Modern meta-heuristics based on nonlinear physics processes: A review of models and design procedures, *Physic Reports* 2016; **655**:1-70.
10. Holland JH. Genetic algorithms, *Sci American* 1992; **267**(1): 66-73.
11. Rechenberg I. *Evolution Strategy: Optimization of Technical systems by means of Biological Evolution*, Fromman-Holzboog, Stuttgart, 1973.
12. 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.
13. Kennedy J, Eberhart R. Particle swarm optimization, *In Proceedings of ICNN'95-International Conference on Neural Networks* 1995; **4:** pp. 1942-1948. IEEE.
14. Karaboga D, *An Idea Based on Honey Bee Swarm for Numerical Optimization*, Technical Report-Tr06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
15. Yang X S, Suash D. Cuckoo Search via Lévy flights, *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)* 2009; pp. 210-214. IEEE.
16. Rashedi E, Nezamabadi-pour H, Saryazdi S. GSA: A gravitational search algorithm, *Inform Sci* 2009; **179**(13): 2232-48.
17. Kaveh A, Talatahari S. A novel heuristic optimization method: charged system search, *Acta Mech* 2010; **213**(3): 267-89.
18. Kaveh A, Akbari H, Hosseini S M. Plasma generation optimization: a new physically-based metaheuristic algorithm for solving constrained optimization problems, *Eng Computat* 2021; **38**(4): 1554-1606.
19. Kaveh A, Hosseini SM, Zaerreza A. A physics-based metaheuristic algorithm based on doppler effect phenomenon and mean euclidian distance threshold, *Period Polytech Civil Eng* 2022 (accepted for publication);
20. Chen VC. *The Micro-Doppler Effect in Radar*, Artech House, United States, 2019.
21. Rosen J, Gothard LQ. *Encyclopedia of Physical Science*, United States, London, 2010.
22. Kaveh A, Hosseini S M, Zaerreza A. Improved Shuffled Jaya algorithm for sizing optimization of skeletal structures with discrete variables, *Struct* 2021; **29**: 107-28.
23. Groenwold A A, Stander N, Snyman J A. A regional genetic algorithm for the discrete optimal design of truss structures, *Int J Numer Meth Eng* 1999; **44**(6): 749-66.
24. Capriles P V S Z, Fonseca L G, Barbosa H J C, Lemonge A C C. Rank-based ant colony algorithms for truss weight minimization with discrete variables, *Communicat Numer Meth Eng* 2007; **23**(6): 553-575.
25. Le D T, Bui D-K, Ngo T D, Nguyen Q-H, Nguyen-Xuan H. A novel hybrid method combining electromagnetism-like mechanism and firefly algorithms for constrained design optimization of discrete truss structures, *Comput Struct* 2019; **212**: 20-42.
26. (AISC) AIOSC. *Manual of Steel Construction Load Resistance Factor Design*, USA, 1994.
27. Hasançebi O, Çarbaş S. Ant colony search method in practical structural optimization, *Int J Optim Civil Eng* 2011; **1**(1): 91-105.
28. Hasançebi O, Azad S K. Adaptive dimensional search: A new metaheuristic algorithm for discrete truss sizing optimization, *Comput Struct* 2015; **154**: 1-16.

29. Hasançebi O, Teke T, Pekcan O. A bat-inspired algorithm for structural optimization, *Comput Struct* 2013; **128**: 77-90.
30. Hasançebi O, Kazemzadeh Azad S. Discrete size optimization of steel trusses using a refined big bang–big crunch algorithm, *Eng Optim* 2014; **46**(1): 61-83.
31. Kaveh A, Mirzaei B, Jafarvand A. Optimal design of double layer barrel vaults using improved magnetic charged system search, *Asian J Civil Eng* 2014; **15**(1): 135-54.
32. Kaveh A, Ghazaan M I. *Meta-heuristic Algorithms for Optimal Design of Real-Size Structures*, Springer, 2018.
33. Kaveh A, Mahdipour Moghanni R, Javadi S M. Optimum design of large steel skeletal structures using chaotic firefly optimization algorithm based on the Gaussian map, *Struct Multidisc Optim* 2019; **60**(3): 879-94.
34. Kaveh A, Zaerreza A, Hosseini S M. An enhanced shuffled Shepherd Optimization Algorithm for optimal design of large-scale space structures, *Eng Comput* 2021.
35. Kaveh A, Zaerreza A, Hosseini S M. Shuffled Shepherd Optimization Method Simplified for Reducing the Parameter Dependency, *Iranian J Sci Technol, Transact Civil Eng* 2021; **45**(3): 1397-1411.