https://doi.org/10.3311/PPci.18670



**Periodica Polytechnica Civil Engineering**, 66(1), pp. 1–17, 2022

# Size/Layout Optimization of Truss Structures Using Vibrating Particles System Meta-heuristic Algorithm and its Improved Version

Ali Kaveh[1*], Masoud Khosravian[1]

[1] School of Civil Engineering, Iran University of Science and Technology, Narmak, Tehran, Postal Code 16846-13114, Iran
[*] Corresponding author, e-mail: alikaveh@iust.ac.ir

Received: 29 May 2021, Accepted: 05 July 2021, Published online: 28 July 2021

**Abstract**

Vibrating Particles System (VPS) optimization is a newly made meta-heuristic algorithm to optimize problems by inspiration of the free vibration of viscous-damped systems with single degree of freedom. The agents are modeled as particles which systematically proceed toward their equilibrium conditions that are reached by the existing population and historically best position. To enhance the performance of the VPS algorithm, Enhanced Vibrating Particles System (EVPS) applies a new process for updating agent's positions. This paper tries to improve the EVPS algorithm with the aim of reduction in the regulatory parameters' effect on the algorithm's performance by reducing the regulatory parameters. To evaluate the performance of the proposed method, it is applied to four optimization problems of truss structures including mixed of discrete and continuous design search spaces with displacement, stress and buckling constraints. As a result, the proposed algorithm is a suitable method and more research can be done on it.

**Keywords**

meta-heuristic algorithms, vibrating particles system algorithm, improved vibrating particles system algorithm, truss optimization, sizing and layout optimization

## 1 Introduction

The term "optimization" means the systematic selection of values for variables from or within a permissible collection, seeking to minimize or maximize the function of a problem [1]. One of the optimization tools is meta-heuristic algorithms that are able to achieve a global or near-global optimal solutions by spending the right amount of time. Some of these algorithms are listed as follows: Particle Swarm Optimization (PSO) [2], Genetic Algorithms (GA) [3], Colliding Bodies Optimization (CBO) [4], Ant Colony Optimization (ACO) [5], Charged System Search algorithm (CSS) [6], Harmony Search (HS) [7], Simulated Annealing (SA) [8], Ray Optimization (RO) [9], Big-Bang Big-Crunch (BBBC) [10], and Vibrating Particles System (VPS) [11].

Researchers in recent years have sought to develop the field of structural optimization by optimizing ideal small structural systems and components to enable the optimal design of more intricate structures [12]. Structural optimization is classified into three kinds: (1) Optimizing the size of sections (2) Optimizing the shape of the structure (3) Optimization of structural topology [13]. Truss structure is one of the most useful civil engineering structures in around the world. Truss optimization has been considered by engineers and researchers for decades [14]. Therefore, truss structures are suitable tool to evaluate the performance of optimization algorithms. Size/layout optimization expresses minimizing the weight of the structure by selecting cross-sections and configuration of the structural members under strength constraints and serviceability limits. Different meta-heuristic algorithms have been used by various researchers to optimize the size and layout of the structure. For instance, Wu and Chow [15] considered the sections and nodal coordinates as a discrete and continuous variables to optimize space trusses by GA algorithm, Hasançebi and Erbatur [16] used "annealing perturbation" and "adaptive reduction of the design space" method to improve GA for mass minimization of space trusses, Hasançebi and Erbatur [17] used Simulated Annealing (SA) algorithm, Kaveh and Kalatjari [18] used the force method and genetic algorithm to minimize the weight of the truss structure, Tang et al. [19] optimized size,

**Cite this article as:** Kaveh, A., Khosravian, M. "Size/Layout Optimization of Truss Structures Using Vibrating Particles System Meta-heuristic Algorithm and its Improved Version", Periodica Polytechnica Civil Engineering, 66(1), pp. 1–17, 2022. https://doi.org/10.3311/PPci.18670

shape and topology of trusses using the improved Genetic Algorithm (GA), Rahami et al. [20] employed energy and force method and Genetic algorithm (GA) for optimize the size and layout of the truss structures, Kazemzadeh Azad et al. [21] suggested a Mutation-Based Real-Coded Genetic Algorithm (MBRCGA) for minimizing the weight of truss, Miguel et al. [22] presented a single-stage Firefly-based Algorithm (FA) to study simultaneous size, shape and topology optimization for trusses, Gholizadeh [23] proposed a hybrid of Cellular Automata (CA) and the Particle Swarm Optimization (PSO) algorithm to solve size and layout optimization of truss structures, Mortazavi et al. [24] utilized integrated Particle Swarm Optimizer (iPSO) algorithm for sizing and shape optimization of planar and spatial truss structures, Panagant and Bureerat [25] suggested Fully Stressed Design-Grey Wolf-Adaptive Differential Evolution (FSD-GWADE), Kaveh and Zaerreza [13] utilized shuffled shepherd optimization algorithm (SSOA), Jawad et al. [26] used Artificial Bee Colony (ABC) algorithm, Kaveh et al. [27] studied layout optimization of planar braced frames by using Colliding Bodies Optimization (CBO) and Colliding Bodies Optimization-Modified Dolphin Monitoring (CBO-MDM).

Vibrating Particles System (VPS) is a multi agent-based randomly optimization algorithm by inspiration of the free vibration of viscous-damped systems with single degree of freedom where particles move toward minimum energy level [11]. The high number of regulatory parameters is one of the features of VPS that affect the performance of the algorithm. Therefore to reduce the impact of these regulatory parameters on the algorithm's performance, in present study, in addition to applying a new process for update the particles position provided by Kaveh et al. [28], by Weighing of regulatory parameters based on the objective function values and adding a mutation mechanism to escape from local optima, standard VPS is improved to formulate the IVPS (Improved Vibrating Particles System) algorithm. The ability of the proposed method are compared to those standard and enhanced version of the vibrating particle system algorithm and some other powerful metaheuristic algorithms on some size/layout optimization of truss structure problems. Results indicate that the proposed method is a promising development to improve the performance of the standard version of the VPS algorithm.

The present paper consists of six sections. After this introduction as the Section 1, in Section 2, the mathematical formulations of sizing-layout truss structural optimization are stated. In Section 3 the vibrating particles system algorithm and its enhanced version is briefly described. The proposed method is explained in Section 4. Four well know sizing and shape optimization of truss structures are studied in Section 5. Finally, conclusions are presented in Section 6.

## 2 Formulation of the structural optimization problems

In the present study, the aim is to optimize size and layout of the truss structures while satisfying under some design constraints to achieve the minimum weight of the structure. Two types of design variables (a) cross sections areas for elements of truss and (b) coordinates of some joints of truss are organized to minimize the weight of truss structures in size and layout optimization. To optimize the size and shape to achieve the lowest weight of the truss structure, the objective function is expressed as follows:

$$\begin{aligned} &\text{Find}\left\{X\right\}=\left[x_1,x_2,\ldots,x_{ng},x_{ng+1},x_{ng+2},\ldots,x_{ng+nn}\right],\\ &\text{To minimize } W\left(\left\{X\right\}\right)=\sum_{i=1}^{nm}\rho_i A_i L_i,\\ &\text{Subjected to}\begin{cases} g_j\left(\left\{X\right\}\right)\le 0, j=1,2,\ldots,nc\\ x_{\min}\le x_i\le x_{\max}\end{cases}. \end{aligned} \tag{1}$$

Where $\{X\}$ is a vector of optimization variables comprising the cross-sectional areas and the coordinates of some nodes; *ng* is the number of member groups for cross-sectional area; *nn* is the number of nodes should be set those coordinates; $W(\{X\})$ presents the weight of the structure; *nm* is the number of elements of the structure; *nc* is the number of constraints; $\rho_i$ presents the material density of *i*th member; $A_i$ and $L_i$ denote the cross-section area and the length of the *i*th member, respectively. $x_{\min}$ and $x_{\max}$ are the minimum and the maximum allowable values of optimization variables in the search space. $g_j(\{X\})$ presents design constraints; and *nc* is the number of constraints.

For handling the different constraints in optimization problems of this paper, a penalty approach is used. So, the objective function (Eq. (1)) is redefined as follows:

$$P\left(\left\{X\right\}\right)=\left(1+\varepsilon_1.v\right)^{\varepsilon_2}\times W\left(\left\{X\right\}\right). \tag{2}$$

Where $P(\{X\})$ presents the penalized objective function that should be minimized, *v* is the sum of the violations of the design constraints. Here, $\varepsilon_1$ is considered as unity and $\varepsilon_2$ is determined by:

$$\varepsilon_2=1.5+1.5\times\frac{iter}{iter_{\max}}, \tag{3}$$

where *iter* is the current iteration number and $iter_{\max}$ is the maximum iteration number.

## 3 Vibrating particles system algorithm

In this section the phenomenon of damped free vibration, the VPS algorithm, the VPS algorithm and the EVPS algorithm are presented briefly.

### 3.1 The damped free vibration phenomenon

Vibration means the reciprocating motion of a particle around its equilibrium position. If the vibration is out of the system due to the presence of periodic force (displacement or velocity), it is called forced vibration, and if it is due to the internal forces of the system, it is called free vibration. A body with mass of $m$ as the vibrating particle, a spring with hardness of $k$ that acts as the inertial force of the system, and a viscous damper with viscous damping coefficient of $c$ that simulates energy loss during vibration (Fig. 1) have been used to modeling the free vibration of viscous-damped systems with single degree of freedom. The motion equation of the body when it shifts as far as $x$ relative to its equilibrium position will be as follows:

$$m\ddot{x} + c\dot{x} + kx = 0 . \tag{4}$$

Two parameters of the damping ratio $\xi$ and the critical damping coefficient $c_c$ are defined to solve this differential equation as follows:

$$\xi = \frac{c}{c_c}, \tag{5}$$

$$c_c = 2m\omega_n , \tag{6}$$

$$\omega_n = \sqrt{\frac{k}{m}} , \tag{7}$$

where $\omega_n$ is the natural circular frequency of the system.

Depending on the value of the damping ratio $\xi$, can be divided into three different types of systems: (1) over-damped system ($\xi > 1$); (2) critically damped system ($\xi = 1$) and (3) under-damped system ($\xi < 1$). A particle only has intermittent motion around its equilibrium position when $\xi < 1$ (under-damped system). In this case, the solution of Eq. (4) is as follows:

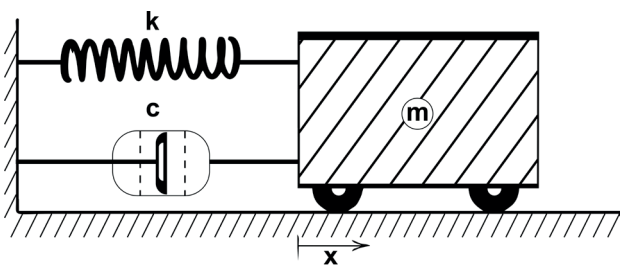


**Fig. 1** Viscous-damped free vibration

$$x(t) = \rho e^{-\xi\omega_n t} \sin\left(\omega_D t + \phi\right), \tag{8}$$

$$\omega_D = \omega_n \sqrt{1-\xi^2} , \tag{9}$$

where $\rho$ and $\phi$ are constants and respect to the initial conditions of the system are calculated. $\omega_D$ and $t$ are damped natural frequency of the system and time, respectively.

### 3.2 The VPS algorithm

In this algorithm, the initial position of the particles is randomly generated in the search space by:

$$x_i^j = x_{\min} + rand.\left(x_{\max} - x_{\min}\right), \tag{10}$$

where $x_i^j$ is the $j$th variable of $i$th particle; *rand* is a random number uniformly distributed in the range of [0, 1].

After calculating the objective function of each particle, three particles HB, GP and BP are used to update the position of the particles, which are the best position from the historically of the whole particle population, good particle and bad particle, respectively. To determine GP and BP, at the first the particle population are sorted in ascending order based on their objective function values, and each of the GP and BP particles is randomly selected from the upper and lower halves, respectively. The following formula is then used to update the particle position:

$$\begin{aligned} x_i^j = {} & \omega_1.\left[D.A.rand1 + HB^j\right] + \omega_2.\left[D.A.rand2 + GP^j\right] \\ & + \omega_3.\left[D.A.rand3 + BP^j\right], \end{aligned} \tag{11}$$

$$A = \left[\omega_1.\left(HB^j - x_i^j\right)\right] + \left[\omega_2.\left(GP^j - x_i^j\right)\right] + \left[\omega_3.\left(BP^j - x_i^j\right)\right], \tag{12}$$

$$D = \left(\frac{iter}{iter_{\max}}\right)^{-\alpha} , \tag{13}$$

$$\omega_1 + \omega_2 + \omega_3 = 1 . \tag{14}$$

Where $A$ and $D$ simulates amplitude of vibration and damping, respectively. $\alpha$ is a constant; $\omega_1$, $\omega_2$, and $\omega_3$ are three parameters to measure the relative importance of HB, GP, and BP, respectively. *rand*1, *rand*2, and *rand*3 are random numbers uniformly distributed in the range of [0,1].

The VPS algorithm is one of the swarm intelligence algorithms that tries to update the position of particles by learning from the existing particle positions. This algorithm obtains the new position of the particles using vibration around three equilibrium positions HB, GP and BP. Moving towards the HB and GP particles provides exploitation phase of the algorithm so that the new

position of the particle approaches the position of the particle with a better objective function. On the other hand, moving towards the BP particle avoids the algorithm being in local optima, and provides exploration phase for the algorithm. Using the combination of these three particles in updating the particle position creates a balance between the exploitation and exploration phases of the algorithm. The probability of the impact of each of the HB, GP and BP particles in updating the new position of particle is controlled using the parameters $\omega_1$, $\omega_2$ and $\omega_3$. Inspired by Eq. (8), Eq. (11) tries to search randomly in the solution space by vibration around three equilibrium positions HB, GP and BP. For this purpose, $A$ (Eq. (12)) simulates the amplitude of the particle oscillation, just like the $\rho$ parameter in Eq. (8), and $D$ (Eq. (13)) simulates the term $e^{-\xi\omega nt}$ in Eq. (8) which represents the damping property by reducing the amplitude of particle motion and causes the answers to converge to an optimal solution, while increasing the iteration loops of the algorithm. Also, in Eq. (11) the value of $\sin(\omega_D t + \phi)$ is considered unity to have vibration with maximum amplitude.

To decide whether the effect of BP should be considered in the updating process of particle position or not, the parameter $p$ is generated between 0 and 1. For this purpose, if $p < rand$, then $\omega_3 = 0$ and $\omega_2 = 1 - \omega_1$ .

To avoid of boundary violation when the particle position is being updated, the harmony search-based side constraint handling approach is used [29]. In this method, the violating component should be changed with the corresponding component of the historically best position of a random particle with the possibility like Harmony Memory Considering Rate (HMCR) parameter, otherwise it must be regenerated from the permissible search space. Moreover, if it is replaced by the component of a historically best position, this value should be changed with the neighboring value by the possibility like Pitch Adjusting Rate (PAR).

### 3.3 The EVPS algorithm

In this section, the Enhanced Vibration Particle System (EVPS) algorithm that presented by Kaveh and Hoseini Vaez [28] is briefly described.

In the EVPS algorithm, A parameter like Memory is defined that it saves NB number of the historically best positions in the whole population. On the other hand, HB is replaced with OHB (one of the historically best positions in the whole population) that it is one row of Memory. The best answer for each iteration is compared with the worst value of the Memory, to be replaced if it is better.

The change in the mechanism of updating the position of the particles is another change made in this method. According to that, one of the (a), (b) and (c) from Eq. (15) is used with the possibility of $\omega_1$, $\omega_2$, and $\omega_3$, respectively instead Eq. (11).

$$x_i^j = \begin{cases} \left[ D.A.rand1 + OHB^j \right];(a) \\ \left[ D.A.rand2 + GP^j \right];(b) \\ \left[ D.A.rand3 + BP^j \right];(c) \end{cases}, \quad A = \begin{cases} \pm 1\left( OHB^j - x_i^j \right);(a) \\ \pm 1\left( GP^j - x_i^j \right);(b) \\ \pm 1\left( BP^j - x_i^j \right);(c) \end{cases} \tag{15}$$

where (±1) are used randomly. OHB, GP and BP are determined independently for each particle.

### 4 Presentation of present method

In this section, the present method is introduced. In the continuation of changes made by Kaveh and Hoseini Vaez [28], corrections will be made to reduce the number of regulatory parameters in the process of the algorithm. These amendments are as follows:

In this method, instead of assigning specific numbers to the parameters $\omega_1$, $\omega_2$, and $\omega_3$, the values are assigned relative to the evaluated objective function of OHB, GP, and BP particles, respectively. Therefore, in order to create a new position for a selected particle ($i$th particle), the following can be done:

Step 1: The initial weights for the four particles OHB, GP, BP and the selected particle are calculated with Eq. (16), Eq. (17), Eq. (18) and Eq. (19), respectively. In these equations, the objective function is considered as a minimization function.

$$m_{OHB} = \frac{1}{Cost_{OHB}}, \tag{16}$$

$$m_{GP} = \frac{1}{Cost_{GP}}, \tag{17}$$

$$m_{BP} = \frac{1}{Cost_{BP}}, \tag{18}$$

$$m_{IP} = \frac{1}{Cost_{IP}}, \tag{19}$$

where $m_{OHB}$, $m_{GP}$, $m_{BP}$ and $m_{IP}$ are the initial weights assigned to the OHB, GP, BP and selected particle,

respectively; $Cost_{OHB}$, $Cost_{GP}$, $Cost_{BP}$ and $Cost_{IP}$ are the objective function values that are calculated for the OHB, GP, BP and selected particle, respectively.

Step 2: According to Eqs. (20) and (21), two key ranks $d_1$ and $d_2$ are defined, respectively. The rank of the selected particle in the list which the particles are sorted in ascending order based on their objective function values (the objective function is considered as a minimization function) is compared by two key ranks $d_1$ and $d_2$. Then, based on the following conditions, the weight of the selected particle ($m_{IP}$) is assigned to the initial weight of one of the three particles OHB, GP and BP:

(a) If the rank of the selected particle in the ordered list is less than $d_1$, according to Eq. (22), the weight of selected particle will be added to the initial weight of the OHB.
(b) If the rank of the selected particle in the ordered list is more than $d_2$, according to Eq. (23), the weight of selected particle will be added to the initial weight of the GP.
(c) Otherwise, according to Eq. (24), the weight of selected particle will be added to the initial weight of the BP.

$$d_1 = \frac{PopulationSize \times \gamma}{4}, \tag{20}$$

$$d_2 = \frac{PopulationSize}{2}, \tag{21}$$

$$m_{OHB}^{new} = (m_{OHB} + m_{IP}) \times \beta \tag{22}$$

$$m_{GP}^{new} = (m_{GP} + m_{IP}) \times \beta \tag{23}$$

$$m_{BP}^{new} = (m_{BP} + m_{IP}) \times \gamma, \tag{24}$$

Where *PopulationSize* is all candidate solutions; $m_{OHB}{}^{new}$, $m_{GP}{}^{new}$ and $m_{BP}{}^{new}$ are the new weights calculated for particles OHB, GP and BP, respectively. To satisfy the balance between the two phases of exploitation and exploration during the increase in number of iterations of the algorithm, we create $\gamma$ and $\beta$ according to Eq. (25) and Eq. (26) as the decremental and incremental parameters, respectively which lead to more exploration-less exploitation in the initial iteration loops of the algorithm and less exploration-more exploitation in the final iteration loops of the algorithm.

$$\gamma = \frac{iter_{\max} - iter}{iter_{\max}} \tag{25}$$

$$\beta = \frac{iter_{\max} + iter}{iter_{\max}} \tag{26}$$

According to the second step, in the initial iteration loops of the algorithm, the effect of good and bad particles decreases and increases, respectively, and in the final iteration loops of the algorithm, the effect of good and bad particles increases and decreases, respectively. Also, by adding the weight of the selected particle ($m_{IP}$) to the weight of the OHB particle ($m_{OHB}$), the convergence speed is increased, and to avoid being in the local optima, we reduce the probability of this effect by considering the $\gamma$ reduction coefficient.

Step 3: Finally, the parameters $\omega_1$, $\omega_2$ and $\omega_3$ which are the three parameters for measuring the relative importance of OHB, GP and BP are equal to the ratio of the new weight of the OHB, GP and BP particles to the total new weight of these three particles, respectively (according to Eq. (27), Eq. (28) and Eq. (29)).

$$\omega_1 = \frac{m_{OHB}^{new}}{m_{OHB}^{new} + m_{GP}^{new} + m_{BP}^{new}}; (\omega_1 \text{ is relative importance to OHB}) \tag{27}$$

$$\omega_2 = \frac{m_{GP}^{new}}{m_{OHB}^{new} + m_{GP}^{new} + m_{BP}^{new}}; (\omega_2 \text{ is relative importance to GP}) \tag{28}$$

$$\omega_3 = \frac{m_{BP}^{new}}{m_{OHB}^{new} + m_{GP}^{new} + m_{BP}^{new}}; (\omega_3 \text{ is relative importance to BP}) \tag{29}$$

Further, in this version of the algorithm, the regulatory parameter $p$ has been removed from the EVPS algorithm. Also, to avoid being in the local optima, a mechanism such as the mutation mechanism in the Genetic Algorithm (GA) and then a new parameter $\mu_0$ has been added to the algorithm. According to Eq. (30), the parameter $\mu$ is obtained based on $\mu_0$ and this parameter is compared with *rand*. If $\mu > rand$, the position of the $j$th variable from the particle according to the Eq. (10) is reproduced within the design range. In the initial iteration loops of the algorithm, $\mu$ parameter is equal to $\mu_0$ and with increasing iterations, this parameter decreases to zero.

$$\mu = \mu_0 \times \gamma, \tag{30}$$

where $\mu_0$ is the mutation rate that is set in the range of [0,1].

## 5 Numerical examples

The performance and applicability of the present method is assessed by four sizing and layout optimization problems and its performance is compared with other versions of VPS algorithm. The parameter of the maximum number of iterations are 500 for the first three problems and 1500 for the

fourth problem. The number of population for all problems is 20. Other parameters for VPS and EVPS algorithms (including $\alpha$, $p$, $\omega_1$, $\omega_2$, NB, HMCR, PAR and neighbor) are set based on [11] and [30], respectively. It should be noted that in the references the value of 0.7 is suggested for the $p$ parameter of VPS algorithm, but in this research, two values of 0.2 and 0.7 have been considered for the mentioned parameter, and the best result for one of these two values is presented in the tables. Also, the MATLAB code provided in [30–32] are used for the EVPS algorithm, the 2017 and 2019 versions of the VPS algorithm, respectively. Thirty independent optimization runs are carried out for each example.

### 5.1 The planar 15-bar truss structure

The 15-bar planar truss structure is the first example to be considered. This truss is affected by a force of 10 kips as shown in Fig. 2. This example consists of 23 variables that 15 of them are in discrete search space for sizing optimization and the rest are in continuous search space for layout optimization. Information about modeling and optimization are presented in Table 1.

Table 2 shows that the present method obtains acceptable results compared to other algorithms. Although MBRCGA [21], SCPSO [23] and iPSO [24] algorithms have better results than the present method, but the best weight obtained by the present method compared to the VPS algorithm and its enhanced version (EVPS) is better, but the number of analyzes that the present method needs to achieve the best weight is more than the other two versions. Fig. 3 shows the best shape for the 15-bar truss obtained by the present method. Fig. 4 shows a comparison between the convergence history of VPS, EVPS algorithms and the present method for the average performance of the algorithms in 30 independent runs. The convergence history of the present method for the mean and best performance of the algorithm in 30 independent runs are shown in Fig. 5.

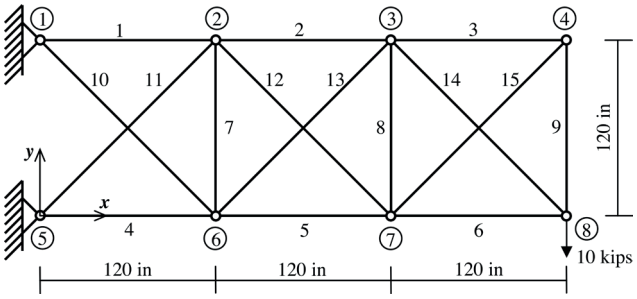


**Fig. 2** Schematic of the 15-bar planar truss

**Table 1** Modeling and design data for the 15-bar planar truss

| Simulation and design data | |
|---|---|
| Design variables: | |
| Sizing variables | $A_i$, $i$ = 1,2,…,15. |
| Layout variables | $x_2 = x_6$; $x_3 = x_7$; $y_2$; $y_3$; $y_4$; $y_6$; $y_7$; $y_8$. |
| Search range: | |
| Possible sizing variables | $A_i \in S$ ={0.111,0.141,0.174,0.220,0.270,0.287, 0.347,0.440,0.539,0.954,1.081,1.174,1.333,1.488, 1.764,2.142,2.697,2.800,3.131,3.565,3.813,4.805, 5.952,6.572,7.192,8.525,9.300,10.850,13.330, 14.290,17.170,19.180}($in^2$). |
| Layout variables bounds | 100 $in.$ ≤ $x_2$ ≤ 140 $in.$; 220 $in.$ ≤ $x_3$ ≤ 260 $in.$; 100 $in.$ ≤ $y_2$ ≤ 140 $in.$; 100 $in.$ ≤ $y_3$ ≤ 140 $in.$; 50 $in.$ ≤ $y_4$ ≤ 90 $in.$; -20 $in.$ ≤ $y_6$ ≤ 20 $in.$; -20 $in.$ ≤ $y_7$ ≤ 20 $in.$; 20 $in.$ ≤ $y_8$ ≤ 60 $in.$ |
| Material Parameters: | |
| Density $\rho$ | 0.1 (lb/$in^3$) |
| Modulus of elasticity $E$ | $10^4$ (ksi) |
| Constraints: | |
| Stress | The allowable elements stress interval: [-25 (ksi), 25 (ksi)] |

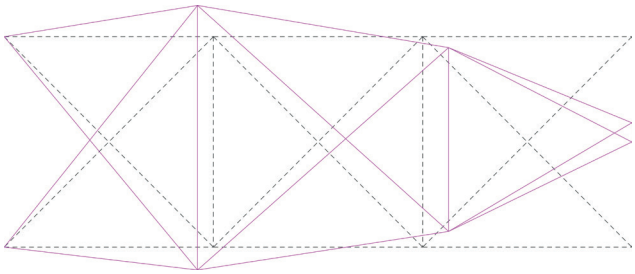

**Fig. 3** Optimum layout of the 15-bar planar truss

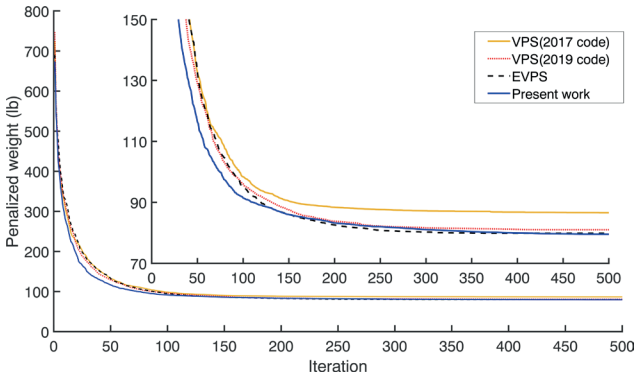


**Fig. 4** Convergence curves for the 15-bar planar truss

### 5.2 The planar 18-bar truss structure

The planar 18-bar truss is at nodes 1, 2, 4, 6 and 8 under vertical loading about 20 kips as shown in Fig. 6. This example has 12 optimization variables that 4 of them are

**Table 2** Performance comparison for the 15-bar planar truss

| Design variables | GA [15] | iGA [19] | MBRCGA [21] | FM-GA [20] | FA [22] | PSO [23] | CPSO [23] | SCPSO [23] | iPSO [24] | VPS (2017 code) $p = 0.2$ | VPS (2019 code) $p = 0.7$ | EVPS $p = 0.2$ | Present work $\mu_0 = 0.03$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A1 | 1.174 | 1.081 | 0.954 | 1.081 | 0.954 | 0.954 | 1.174 | 0.954 | 1.081 | 0.954 | 0.954 | 1.081 | 0.954 |
| A2 | 0.954 | 0.539 | 0.539 | 0.539 | 0.539 | 1.081 | 0.539 | 0.539 | 0.539 | 0.954 | 0.539 | 0.539 | 0.539 |
| A3 | 0.44 | 0.287 | 0.111 | 0.287 | 0.22 | 0.27 | 0.347 | 0.27 | 0.27 | 0.27 | 0.27 | 0.287 | 0.22 |
| A4 | 1.333 | 0.954 | 0.954 | 0.954 | 0.954 | 1.081 | 0.954 | 0.954 | 0.954 | 1.081 | 0.954 | 0.954 | 0.954 |
| A5 | 0.954 | 0.954 | 0.539 | 0.539 | 0.539 | 0.539 | 0.954 | 0.539 | 0.539 | 0.539 | 0.539 | 0.539 | 0.539 |
| A6 | 0.174 | 0.22 | 0.347 | 0.141 | 0.22 | 0.287 | 0.141 | 0.174 | 0.141 | 0.22 | 0.22 | 0.141 | 0.27 |
| A7 | 0.44 | 0.111 | 0.111 | 0.111 | 0.111 | 0.141 | 0.141 | 0.111 | 0.111 | 0.111 | 0.111 | 0.111 | 0.111 |
| A8 | 0.44 | 0.111 | 0.111 | 0.111 | 0.111 | 0.111 | 0.111 | 0.111 | 0.111 | 0.141 | 0.111 | 0.111 | 0.111 |
| A9 | 1.081 | 0.287 | 0.111 | 0.539 | 0.287 | 0.347 | 1.174 | 0.287 | 0.27 | 1.488 | 0.954 | 0.287 | 0.174 |
| A10 | 1.333 | 0.22 | 0.44 | 0.44 | 0.44 | 0.44 | 0.141 | 0.347 | 0.287 | 0.347 | 0.44 | 0.347 | 0.44 |
| A11 | 0.174 | 0.44 | 0.44 | 0.539 | 0.44 | 0.27 | 0.44 | 0.347 | 0.44 | 0.22 | 0.44 | 0.44 | 0.44 |
| A12 | 0.174 | 0.44 | 0.174 | 0.27 | 0.22 | 0.111 | 0.44 | 0.22 | 0.27 | 0.174 | 0.141 | 0.22 | 0.174 |
| A13 | 0.347 | 0.111 | 0.174 | 0.22 | 0.22 | 0.347 | 0.141 | 0.22 | 0.287 | 0.44 | 0.111 | 0.27 | 0.174 |
| A14 | 0.347 | 0.22 | 0.347 | 0.141 | 0.27 | 0.44 | 0.141 | 0.174 | 0.174 | 0.22 | 0.22 | 0.141 | 0.27 |
| A15 | 0.44 | 0.347 | 0.111 | 0.287 | 0.22 | 0.22 | 0.347 | 0.27 | 0.27 | 0.27 | 0.27 | 0.287 | 0.22 |
| x2 | 123.189 | 133.612 | 105.7835 | 101.5775 | 114.967 | 106.0521 | 102.2873 | 137.2216 | 132.2415 | 129.1251 | 139.7080 | 116.1486 | 110.8864 |
| x3 | 231.595 | 234.752 | 258.5965 | 227.9112 | 247.04 | 239.0245 | 240.505 | 259.9093 | 257.4379 | 247.3116 | 257.5795 | 251.4127 | 254.8960 |
| y2 | 107.189 | 100.449 | 133.6284 | 134.7986 | 125.919 | 130.3556 | 112.584 | 123.5006 | 128.3136 | 116.4325 | 133.1802 | 128.9126 | 137.8297 |
| y3 | 119.175 | 104.738 | 105.0023 | 128.2206 | 111.067 | 114.273 | 108.0428 | 110.002 | 111.2506 | 104.9919 | 102.8041 | 109.2586 | 113.7671 |
| y4 | 60.462 | 73.762 | 54.4546 | 54.863 | 58.298 | 51.9866 | 57.7952 | 59.9356 | 59.9894 | 59.4382 | 56.8942 | 55.7240 | 70.9138 |
| y6 | -16.728 | -10.067 | -19.929 | -16.4484 | -17.564 | 1.8135 | 6.4299 | -5.1799 | -10.5543 | -0.2914 | -11.2360 | -12.9730 | -13.0542 |
| y7 | 15.565 | -1.339 | 3.6223 | -13.301 | -5.821 | 9.1827 | -1.8006 | 4.2193 | 10.7686 | -0.6786 | 10.1297 | -9.0125 | 8.9559 |
| y8 | 36.645 | 50.402 | 54.4474 | 54.8572 | 31.465 | 46.9087 | 57.7987 | 57.8829 | 60 | 59.4382 | 56.8942 | 55.7239 | 59.9658 |
| Best weight (lb) | 120.52 | 79.82 | 72.5152 | 76.6854 | 75.55 | 82.2344 | 77.6153 | 72.5143 | 72.7373 | 78.7876 | 74.5743 | 74.8172 | 73.7463 |
| Average weight (lb) | N/A | N/A | N/A | N/A | N/A | N/A | N/A | 76.411 | 74.316 | 86.6161 | 81.0198 | 79.8588 | 79.5491 |
| Worst weight (lb) | N/A | N/A | N/A | N/A | N/A | N/A | N/A | 80.156 | 76.522 | 99.4843 | 89.6244 | 85.4333 | 85.7443 |
| Std. deviation (lb) | N/A | N/A | N/A | N/A | N/A | N/A | N/A | 1.922 | 2.023 | 4.66 | 3.11 | 2.32 | 2.15 |
| No. of analyses | 6,000 | 8,000 | 10,000 | 8,000 | 8,000 | 4,500 | 4,500 | 4,500 | 4,980 | 8,840 | 7,860 | 9,820 | 9,940 |

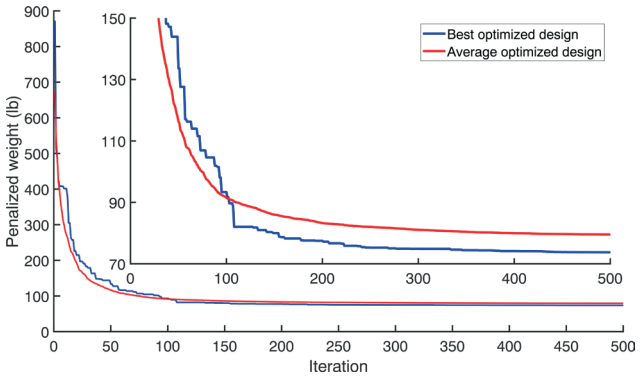


**Fig. 5** Convergence curves of the best and average performance of the studied algorithm for the 15-bar planar truss

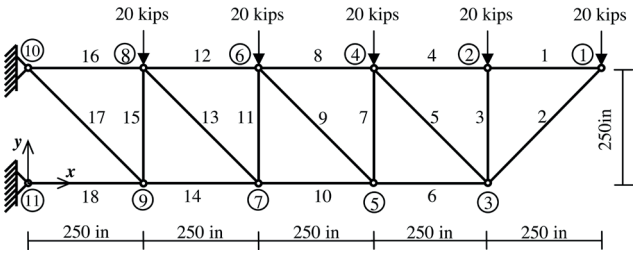


**Fig. 6** Schematic of the 18-bar planar truss

in discrete search space which are related to member sections and 8 of them are in continuous search space that are related to node coordinates. Information about modeling and optimization are presented in Table 3.

Table 4 shows that, like the 15-bar truss results, the MBRCGA [21], SCPSO [23] and iPSO [24] algorithms performed better than the present method, but still the best weight obtained by the present method (i.e., 4,525.09 lbs) is better than the values obtained by the VPS algorithm and its enhanced version (EVPS). In addition to the fact that the present method requires more analysis to achieve the best weight than the other two versions of the VPS algorithm, the average and standard deviation for 30 independent runs were also more values. Optimum layout for 18-bar truss is shown in Fig. 7 that found by the present method. Fig. 8 shows a comparison between the convergence history of VPS, EVPS algorithms and the present method for the average performance of the algorithms in 30 independent runs. The convergence history of the present method for the mean and best performance of the algorithm in 30 independent runs are shown in Fig. 9.

### 5.3 The spatial 25-bar truss

As shown in Fig. 10 the 25-bar truss is considered as the third problem. This structure is under concentrated loading as shown in Table 5. This example consists of 13

**Table 3** Modeling and design data for the 18-bar planar truss

| Simulation and design data | |
|---|---|
| Design variables: | |
| Sizing variables | $A_1 = A_4 = A_8 = A_{12} = A_{16}$, $A_2 = A_6 = A_{10} = A_{14} = A_{18}$, $A_3 = A_7 = A_{11} = A_{15}$, $A_5 = A_9 = A_{13} = A_{17}$. |
| Layout variables | $x_3, x_5, x_7, x_9, y_3, y_5, y_7, y_9$. |
| Search range: | |
| Possible sizing variables | $A_i \in S$ ={2.00, 2.25, 2.50, …, 21.25, 21.50, 21.75}($in^2$) |
| Layout variables bounds | 775 $in.$ ≤ $x_3$ ≤ 1225 $in.$; 525 $in.$ ≤ $x_5$ ≤ 975 $in.$; 275 $in.$ ≤ $x_7$ ≤ 725 $in.$; 25 $in.$ ≤ $x_9$ ≤ 475 $in.$; -225 $in.$ ≤ $y_3, y_5, y_7, y_9$ ≤ 245 $in.$ |
| Material Parameters: | |
| Density $\rho$ | 0.1 (lb/$in^3$) |
| Modulus of elasticity $E$ | $10^4$ (ksi) |
| Constraints: | |
| Stress | The allowable elements stress interval: [-25 (ksi), 25 (ksi)] |
| Local buckling | $\left\|(\sigma_c)_i\right\| \le \frac{\beta E A_i}{l_i^2}$, $i = 1,2,\ldots,18$, $\beta = 4$ |

variables that 8 of them are in discrete search space for sizing optimization and the rest are in continuous search space for layout optimization. Other information related to modeling and optimization are given in Table 5.

According to the results shown in Table 6, VPS algorithm, SCPSO [23] and iPSO [24] found a smaller weight as the best weight, which had better results than the present method, while the present method has achieved less weight compared to the EVPS algorithm. Comparison between the optimum layout obtained by the present method and initial layout is shown in Fig. 11. Comparison of the average performance of VPS, EVPS algorithms and the present method for 30 independent runs are shown in Fig. 12. The convergence history of the present method for the mean and best performance of the algorithm in 30 independent runs are shown in Fig. 13.

### 5.4 The planar 47-bar truss

The last sizing and shape optimization problem is the planar 47-bar truss that is indicated in Fig. 14 which consist 47 members and 22 nodes. The nodes of structure are subjected to three load cases that are presented in Table 7. This example has 44 optimization variables that 27 of them are in discrete search space which are related to member sections and 17 of them are in continuous search space that are

**Table 4** Performance comparison for the 18-bar planar truss

| Design variables | SA [17] | FM-GA [18] | MBRCGA [21] | FM-GA [20] | PSO [23] | CPSO [23] | SCPSO [23] | iPSO [24] | VPS (2017 code) $p = 0.2$ | VPS (2019 code) $p = 0.2$ | EVPS $p = 0.2$ | Present Work $\mu_0 = 0.03$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A1 | 12.25 | 12.25 | 12.75 | 12.75 | 12 | 12 | 12.5 | 14.25 | 13 | 11.75 | 12.25 | 12 |
| A2 | 17.5 | 18 | 18.25 | 18.5 | 18.5 | 17.25 | 17.5 | 11.75 | 16.75 | 18.25 | 18 | 18 |
| A3 | 5.75 | 5.25 | 5 | 4.75 | 5.25 | 6.25 | 5.75 | 6 | 7.25 | 5.25 | 5.25 | 5 |
| A5 | 4.25 | 4.25 | 3.25 | 3.25 | 4.5 | 4.75 | 3.75 | 8 | 3.5 | 5 | 4.25 | 4.5 |
| x3 | 910 | 913 | 916.0812 | 917.4475 | 903.9806 | 902.9141 | 907.2491 | 916.4975 | 895.9864 | 930.8557 | 916.2911 | 918.1752 |
| y3 | 179 | 186.8 | 191.43 | 193.7899 | 185.7807 | 174.7201 | 179.8671 | 190.5241 | 162.2994 | 188.6362 | 189.8249 | 190.8539 |
| x5 | 638 | 650 | 650.0573 | 654.3243 | 644.917 | 632.7129 | 636.7873 | 916.4975 | 616.1673 | 661.1871 | 650.3419 | 651.8908 |
| y5 | 141 | 150.5 | 153.4968 | 159.9436 | 144.9692 | 141.2956 | 141.8271 | 152.9217 | 124.1998 | 147.9244 | 150.7933 | 149.8333 |
| x7 | 408 | 418.8 | 419.4508 | 424.4821 | 428.2196 | 407.1323 | 407.9442 | 649.4695 | 390.8926 | 422.3898 | 419.3627 | 420.0147 |
| y7 | 91 | 97.4 | 105.5322 | 108.5779 | 100.5623 | 85.9332 | 94.0559 | 105.425 | 86.0397 | 93.9783 | 98.9555 | 97.9757 |
| x9 | 198 | 204.8 | 205.6591 | 208.4691 | 209.5415 | 197.672 | 198.7897 | 205.4255 | 188.8596 | 209.3734 | 205.0683 | 205.5626 |
| y9 | 24 | 26.7 | 36.4848 | 37.6349 | 24.3748 | 19.8093 | 29.5157 | 36.4252 | 33.9445 | 18.5291 | 27.5897 | 23.1446 |
| Best weight (lb) | 4533.24 | 4547.9 | 4520.2 | 4530.7 | 4609.001 | 4561.131 | 4512.365 | 4520.99 | 4,595.53 | 4,600.02 | 4,544.25 | 4,525.09 |
| Average weight (lb) | N/A | N/A | N/A | N/A | N/A | N/A | 4551.709 | 4526.585 | 4,978.03 | 4,769.76 | 4,743.40 | 4,798.26 |
| Worst weight (lb) | N/A | N/A | N/A | N/A | N/A | N/A | 4621.227 | 4560.27 | 5,747.31 | 4,993.75 | 5,112.79 | 5,147.76 |
| Std. deviation (lb) | N/A | N/A | N/A | N/A | N/A | N/A | 37.691 | 14.889 | 244.24 | 89.16 | 133.33 | 169.86 |
| No. of analyses | N/A | N/A | 10,000 | N/A | 4,500 | 4,500 | 4,500 | 4,450 | 8,300 | 9,780 | 9,400 | 10,000 |

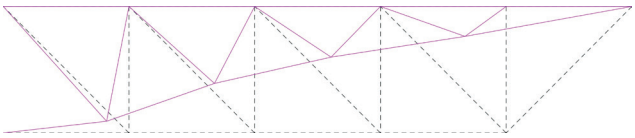

**Fig. 7** Optimum layout of the 18-bar planar truss

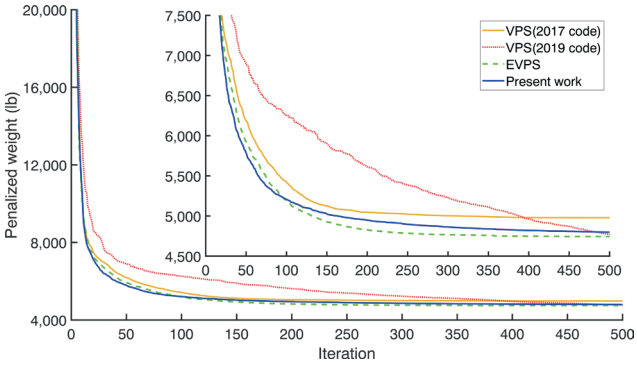


**Fig. 8** Convergence curves for the 18-bar planar truss

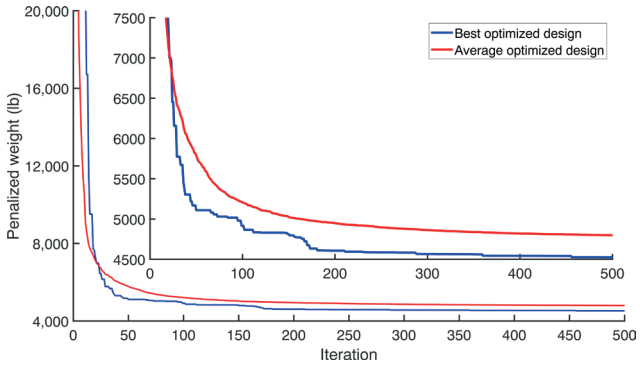


**Fig. 9** Convergence curves of the best and average performance of the studied algorithm for the 18-bar planar truss

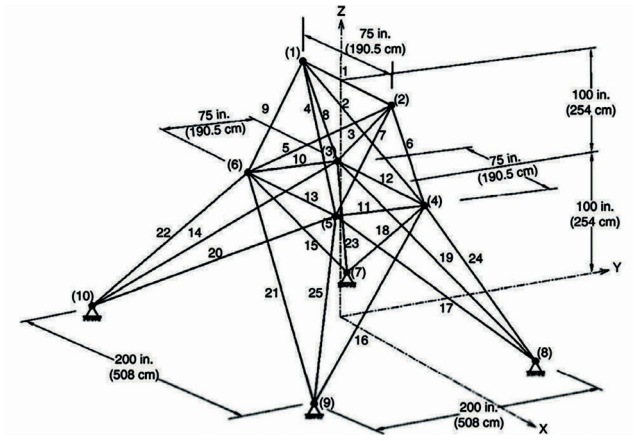


**Fig. 10** Schematic of the 25-bar spatial truss

related to node coordinates. Other information for modeling and optimization are given in Table 7.

According to the results of Table 8, the SA [17], FSD-GWADE [25], CPSO [23], SCPSO [23] and iPSO [24] algorithms have better results than the present method, but the best and average weights for 30 independent runs found by the present method is better than VPS and EVPS algorithms, but the number of analyzes is more. Comparison between the optimum layout obtained by the present method and initial layout is shown in Fig. 15. Fig. 16 shows the comparison of convergence histories of the 47-bar truss structure for VPS, EVPS and the present method. Fig. 17 is compared the convergence histories of average and best performance for the present method.

**Table 5** Modeling and design data for the 25-bar spatial truss

| Simulation and design data | | | |
|---|---|---|---|
| design variables: | | | |
| Sizing variables | $A_1; A_2 = A_3 = A_4 = A_5; A_6 = A_7 = A_8 = A_9;$ $A_{10} = A_{11}; A_{12} = A_{13}; A_{14} = A_{15} = A_{16} = A_{17};$ $A_{18} = A_{19} = A_{20} = A_{21}; A_{22} = A_{23} = A_{24} = A_{25};$ | | |
| Layout variables | $x_4 = x_5 = -x_3 = -x_6; x_8 = x_9 = -x_7 = -x_{10};$ $y_3 = y_4 = -y_5 = -y_6; y_7 = y_8 = -y_9 = -y_{10};$ $z_3 = z_4 = z_5 = z_6;$ | | |
| Search range: | | | |
| Possible sizing variables | $A_i \in S = \{0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0,1.1,$ $1.2,1.3,1.4,1.5,1.6,1.7,1.8,1.9,2.0,2.1,2.2,2.3,2.4,2.5,$ $2.6,2.8,3.0,3.2,3.4\}(in^2)$. | | |
| Layout variables bounds | $20\ in. \le x_4 \le 60\ in.; 40\ in. \le x_8 \le 80\ in.;$ $40\ in. \le y_4 \le 80\ in.; 100\ in. \le y_8 \le 140\ in.;$ $90\ in. \le z_4 \le 130\ in.$ | | |
| Material Parameters: | | | |
| Density $\rho$ | 0.1 (lb/$in^3$) | | |
| Modulus of elasticity $E$ | $10^4$ (ksi) | | |
| Constraints: | | | |
| Stress | The allowable elements stress interval: [-40 (ksi), 40 (ksi)] | | |
| Displacement | The allowable nodal displacement interval: [-0.35 (in.), 0.35 (in.)] | | |
| Loads: | | | |
| Nodes | $F_x$(kips) | $F_y$(kips) | $F_z$(kips) |
| 1 | 1.0 | -10.0 | -10.0 |
| 2 | 0.0 | -10.0 | -10.0 |
| 3 | 0.5 | 0.0 | 0.0 |
| 6 | 0.6 | 0.0 | 0.0 |

## 6 Conclusions

In present paper, the ability of the new enhanced version of VPS metaheuristic algorithm that is named Improved Vibrating Particles System (IVPS) algorithm in optimizing size/layout the truss structures is investigated. VPS is a multi agent-based algorithm by inspiration of the free vibration of viscous-damped systems with single degree of freedom. The present method has tried to reduce the dependence of the VPS algorithm on its regulatory parameters by weighting the regulatory parameters of the VPS algorithm using the objective function values.

**Table 6** Performance comparison for the 25-bar spatial truss

| Design variables | GA [15] | iGA [19] | FM-GA [18] | FA [22] | FM-GA [20] | PSO [23] | CPSO [23] | SCPSO [23] | iPSO [24] | VPS (2017 code) $p = 0.2$ | VPS (2019 code) $p = 0.2$ | EVPS $p = 0.2$ | Present Work $\mu_0 = 0.03$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.3 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| A2 | 0.2 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| A6 | 1.1 | 1.1 | 1.1 | 0.9 | 1.1 | 1.1 | 1 | 1 | 1 | 1 | 1 | 0.9 | 1 |
| A10 | 0.2 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| A12 | 0.3 | 0.1 | 0.1 | 0.1 | 0.1 | 0.4 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| A14 | 0.1 | 0.2 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| A18 | 0.2 | 0.2 | 0.1 | 0.1 | 0.2 | 0.4 | 0.2 | 0.1 | 0.1 | 0.1 | 0.1 | 0.2 | 0.1 |
| A22 | 0.9 | 0.7 | 1 | 1 | 0.8 | 0.7 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 |
| x4 | 41.07 | 35.47 | 36.23 | 37.32 | 33.0487 | 27.6169 | 33.4976 | 36.952 | 37.6 | 37.6171 | 37.6475 | 32.6972 | 37.5279 |
| y4 | 53.47 | 60.37 | 58.56 | 55.74 | 53.5663 | 51.6196 | 62.3735 | 54.579 | 54.46 | 54.4361 | 54.4454 | 53.4719 | 54.8148 |
| z4 | 124.6 | 129.07 | 115.59 | 126.62 | 129.9092 | 129.9071 | 114.5945 | 129.976 | 130 | 130.0000 | 129.9998 | 129.4781 | 129.4257 |
| x8 | 50.8 | 45.06 | 46.46 | 50.14 | 43.7826 | 42.5526 | 40.0531 | 51.732 | 51.89 | 51.8914 | 51.8893 | 43.2422 | 51.7433 |
| y8 | 131.48 | 137.04 | 127.95 | 136.4 | 136.8381 | 132.7241 | 133.6695 | 139.532 | 139.55 | 139.5491 | 139.5388 | 137.4310 | 139.5783 |
| Best weight (lb) | 136.2 | 124.94 | 124 | 118.83 | 120.1149 | 129.2076 | 123.451 | 117.227 | 117.255 | 117.2556 | 117.2556 | 119.3020 | 117.2900 |
| Average weight (lb) | N/A | N/A | N/A | N/A | N/A | N/A | N/A | 122.876 | 119.57 | 123.4255 | 118.6200 | 124.1990 | 121.8993 |
| Worst weight (lb) | N/A | N/A | N/A | N/A | N/A | N/A | N/A | 132.672 | 121.969 | 144.7848 | 122.4392 | 137.6314 | 126.8062 |
| Std. deviation (lb) | N/A | N/A | N/A | N/A | N/A | N/A | N/A | 3.671 | 1.3908 | 5.62 | 1.34 | 3.92 | 2.32 |
| No. of analyses | N/A | 6,000 | N/A | 6,000 | 10,000 | 4,500 | 4,500 | 4500 | 4870 | 5,840 | 9,360 | 5,080 | 9,120 |

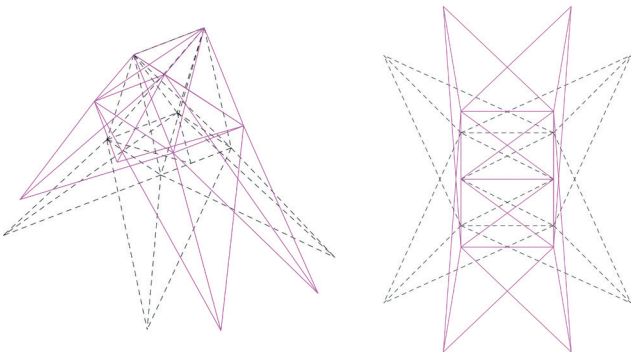

**Fig. 11** Optimum layout of the 25-bar spatial truss

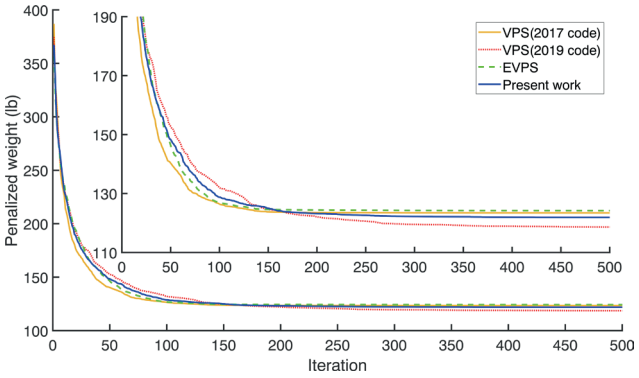


**Fig. 12** Convergence curves for the 25-bar spatial truss

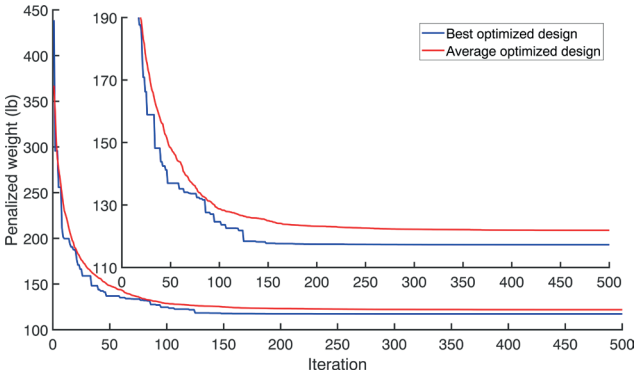


**Fig. 13** Convergence curves of the best and average performance of the studied algorithm for the 25-bar spatial truss

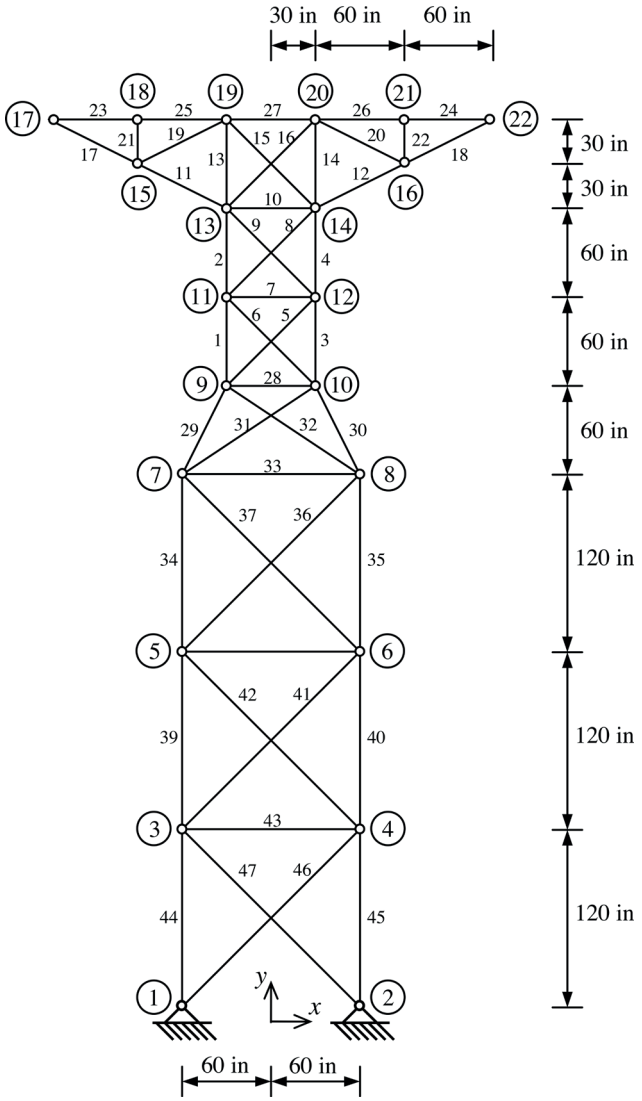


**Fig. 14** Schematic of the 47-bar planar truss

All of the optimization problems of truss structure in this paper involve size and layout variables and the fourth problem with 44 optimization variables is a slightly larger scale problem. The results of the present method in all problems were close to the best results obtained by other powerful algorithms and in some cases, it achieved better results than them. Also, in all structures except 25-bar spatial truss, it has achieved better results than other versions of the vibrating particle system algorithm. The weight gained by the standard version of the vibrating particle system algorithm was slightly different (0.029%) from the current method for the spatial 25-bar truss. In this paper, only four small size/layout optimization truss structures have been used to evaluate the capability of present method, so the comparison between the performance of the present method and other versions of the vibrating particle system algorithm can be tested using other benchmark structures.

## Compliance with ethical standards

Conflict of interest: No potential conflict of interest was reported by the authors.

**Table 7** Modeling and design data for the 47-bar planar truss

| Simulation and design data | | | |
|---|---|---|---|
| Design variables: | | | |
| Sizing variables | $A_1 = A_3$; $A_2 = A_4$; $A_5 = A_6$; $A_7$; $A_8 = A_9$; $A_{10}$; $A_{11} = A_{12}$; $A_{13} = A_{14}$; $A_{15} = A_{16}$; $A_{17} = A_{18}$; $A_{19} = A_{20}$; $A_{21} = A_{22}$; $A_{23} = A_{24}$; $A_{25} = A_{26}$; $A_{27}$; $A_{28}$; $A_{29} = A_{30}$; $A_{31} = A_{32}$; $A_{33}$; $A_{34} = A_{35}$; $A_{36} = A_{37}$; $A_{38}$; $A_{39} = A_{40}$; $A_{41} = A_{42}$; $A_{43}$; $A_{44} = A_{45}$; $A_{46} = A_{47}$. | | |
| Layout variables | $x_1 = -x_2$; $x_3 = -x_4$; $y_3 = y_4$; $x_5 = -x_6$; $y_5 = y_6$; $x_7 = -x_8$; $y_7 = y_8$; $x_9 = -x_{10}$; $y_9 = y_{10}$; $x_{11} = -x_{12}$; $y_{11} = y_{12}$; $x_{13} = -x_{14}$; $y_{13} = y_{14}$; $x_{19} = -x_{20}$; $y_{19} = y_{20}$; $x_{18} = -x_{21}$; $y_{18} = y_{21}$. | | |
| Search range: | | | |
| Possible sizing variables | $A_i \in S = \{0.1, 0.2, 0.3, 0.4, \ldots, 4.8, 4.9, 5.0\}(in^2)$. | | |
| Layout variables bounds | $0\ in. \le x_i \le 120\ in.$, ($i$ =2, 4, 6, 8); $-30\ in. \le x_i \le 90\ in.$, ($i$ = 10, 12, 14, 20); $30\ in. \le x_{21} \le 150\ in.$; $60\ in. \le y_4 \le 180\ in.$; $180\ in. \le y_6 \le 300\ in.$; $300\ in. \le y_8 \le 420\ in.$; $360\ in. \le y_{10} \le 480\ in.$; $420\ in. \le y_{12} \le 540\ in.$; $480\ in. \le y_{14} \le 600\ in.$; $540\ in. \le y_{20} \le 660\ in.$; $540\ in. \le y_{21} \le 660\ in.$ | | |
| Material Parameters: | | | |
| Density $\rho$ | 0.3 (lb/$in^3$) | | |
| Modulus of elasticity $E$ | $3 \times 10^4$ (ksi) | | |
| Constraints: | | | |
| Stress | The allowable elements stress interval: [-15 (ksi), 20 (ksi)] | | |
| Local buckling | $\left\|(\sigma_c)_i\right\| \le \frac{\beta E A_i}{l_i^2}$, $i = 1, 2, \ldots, 47$, $\beta = 3.96$. | | |
| Loads: | | | |
| case | Nodes | $F_x$(kips) | $F_y$(kips) |
| 1 | 17 | 6.0 | -14.0 |
| | 22 | 6.0 | -14.0 |
| 2 | 17 | 6.0 | -14.0 |
| 3 | 22 | 6.0 | -14.0 |

**Table 8** Performance comparison for the 47-bar planar truss

| Design variables | IGA [16] | SA [17] | FSD-GWADE [25] | PSO [23] | CPSO [23] | SCPSO [23] | iPSO [24] | VPS (2017 code) $p = 0.2$ | VPS (2019 code) $p = 0.2$ | EVPS $p = 0.2$ | Present Work $\mu_0 = 0.02$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| A3 | 2.5 | 2.5 | 2.7 | 2.8 | 2.6 | 2.5 | 2.5 | 0.6 | 4.5 | 2.7 | 2.9 |
| A4 | 2.2 | 2.5 | 2.6 | 2.7 | 2.5 | 2.5 | 2.5 | 0.9 | 3 | 2.4 | 2.6 |
| A5 | 0.7 | 0.8 | 0.7 | 0.8 | 0.7 | 0.8 | 0.8 | 3.8 | 1 | 0.8 | 0.8 |
| A7 | 0.1 | 0.1 | 0.1 | 1.1 | 0.3 | 0.1 | 0.1 | 0.1 | 0.2 | 0.3 | 0.1 |
| A8 | 1.3 | 0.7 | 0.8 | 0.8 | 1.2 | 0.7 | 0.7 | 3.8 | 1.2 | 1.1 | 1.3 |
| A10 | 1.3 | 1.3 | 1.2 | 1.3 | 1.1 | 1.4 | 1.4 | 1.3 | 1 | 1.3 | 1 |
| A12 | 1.8 | 1.8 | 1.7 | 1.8 | 1.6 | 1.7 | 1.7 | 2.1 | 2 | 2 | 1.9 |
| A14 | 0.5 | 0.7 | 0.8 | 0.9 | 0.8 | 0.8 | 0.8 | 0.9 | 1.7 | 0.5 | 0.6 |
| A15 | 0.8 | 0.9 | 0.9 | 1.2 | 1.1 | 0.9 | 0.9 | 1.6 | 0.9 | 0.8 | 0.8 |
| A18 | 1.2 | 1.2 | 1.3 | 1.4 | 1.3 | 1.3 | 1.3 | 1.5 | 2.4 | 1.7 | 1.4 |
| A20 | 0.4 | 0.4 | 0.3 | 0.3 | 0.3 | 0.3 | 0.3 | 0.9 | 0.6 | 0.6 | 0.3 |
| A22 | 1.2 | 1.3 | 1 | 1.4 | 0.8 | 0.9 | 0.9 | 1.4 | 1.4 | 0.9 | 1 |
| A24 | 0.9 | 0.9 | 1 | 1.1 | 1 | 1 | 1 | 1.1 | 1 | 1.3 | 1.1 |
| A26 | 1 | 0.9 | 1 | 1.2 | 1 | 1.1 | 1.1 | 1 | 0.7 | 1.4 | 1.1 |
| A27 | 3.6 | 0.7 | 0.9 | 1.6 | 0.9 | 5 | 0.9 | 1.4 | 0.9 | 1.2 | 0.9 |
| A28 | 0.1 | 0.1 | 0.1 | 1 | 0.1 | 0.1 | 0.1 | 0.2 | 0.4 | 0.3 | 0.1 |
| A30 | 2.4 | 2.5 | 2.6 | 2.8 | 2.7 | 2.5 | 2.5 | 3.3 | 3 | 2.6 | 2.6 |
| A31 | 1.1 | 1 | 0.9 | 0.8 | 0.9 | 1 | 1 | 1 | 1.1 | 1 | 0.9 |
| A33 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 1.5 | 0.3 | 0.1 | 0.1 |
| A35 | 2.7 | 2.9 | 2.8 | 3 | 3 | 2.8 | 2.8 | 3.3 | 3.2 | 2.6 | 2.8 |
| A36 | 0.8 | 0.8 | 1.1 | 0.9 | 1 | 0.9 | 0.9 | 1.2 | 1.2 | 1.1 | 1 |
| A38 | 0.1 | 0.1 | 0.1 | 0.1 | 0.2 | 0.1 | 0.1 | 1.7 | 0.1 | 0.2 | 0.1 |
| A40 | 2.8 | 3 | 3 | 3.3 | 3.3 | 3 | 3 | 3.6 | 3.2 | 2.7 | 3.1 |
| A41 | 1.3 | 1.2 | 1.1 | 0.9 | 0.9 | 1 | 1 | 1.6 | 1.4 | 1.3 | 1 |
| A43 | 0.2 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 | 0.9 | 0.3 | 0.1 | 0.1 |
| A45 | 3 | 3.2 | 3.1 | 3.3 | 3.3 | 3.2 | 3.2 | 4.5 | 3.3 | 3 | 3.3 |
| A46 | 1.2 | 1.1 | 1.1 | 1.2 | 1.1 | 1.2 | 1.2 | 0.8 | 1.3 | 1.2 | 1.3 |
| x2 | 114 | 104 | 109.61 | 98.8628 | 99.363 | 101.3393 | 101.2077 | 76.8274 | 102.9225 | 115.9306 | 95.5386 |
| x4 | 97 | 87 | 93.078 | 78.6595 | 83.4439 | 85.9111 | 85.8555 | 67.3946 | 78.7688 | 103.3133 | 80.5830 |
| y4 | 125 | 128 | 126.65 | 146.7331 | 126.3845 | 135.9645 | 135.9679 | 86.1660 | 145.3218 | 114.5742 | 142.8836 |
| x6 | 76 | 70 | 70.752 | 66.5231 | 69.5148 | 74.7969 | 74.9087 | 62.8919 | 64.4627 | 84.3044 | 69.8276 |
| y6 | 261 | 259 | 246.32 | 239.0901 | 218.2013 | 237.7447 | 238.0442 | 215.7321 | 271.2362 | 255.6127 | 253.0915 |

Continuing Table 8

| Design variables | IGA [16] | SA [17] | FSD-GWADE [25] | PSO [23] | CPSO [23] | SCPSO [23] | iPSO [24] | VPS (2017 code) $p = 0.2$ | VPS (2019 code) $p = 0.2$ | EVPS $p = 0.2$ | Present Work $\mu_0 = 0.02$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| x8 | 69 | 62 | 56.172 | 55.6936 | 58.0004 | 64.3115 | 64.1206 | 47.7076 | 45.9873 | 64.5654 | 57.6594 |
| y8 | 316 | 326 | 356.26 | 327.7882 | 322.2272 | 321.3416 | 321.5037 | 343.3780 | 364.9041 | 359.9130 | 347.4688 |
| x10 | 56 | 53 | 48.498 | 48.8641 | 51.4015 | 53.3345 | 53.3481 | 34.4688 | 42.6528 | 45.7518 | 49.0938 |
| y10 | 414 | 412 | 436.37 | 398.6775 | 401.5626 | 414.3025 | 413.7265 | 419.6109 | 443.0305 | 432.0751 | 436.6117 |
| x12 | 50 | 47 | 42.37 | 43.14 | 46.8605 | 46.0277 | 46.2881 | -28.2689 | 43.9389 | 41.9274 | 37.0847 |
| y12 | 463 | 486 | 490.66 | 464.7831 | 458.3021 | 489.9216 | 487.9695 | 465.6604 | 457.9637 | 486.7322 | 483.2620 |
| x14 | 54 | 45 | 41.61 | 37.8993 | 46.8885 | 41.8353 | 41.8603 | 30.6771 | 49.3600 | 49.9330 | 47.8346 |
| y14 | 524 | 504 | 521.04 | 511.045 | 527.8575 | 522.4161 | 522.8897 | 505.6582 | 518.5287 | 526.3668 | 524.8298 |
| x20 | 1 | 2 | 1.4026 | 18.2341 | 16.2354 | 1.0005 | 0.9892 | 4.6435 | -10.8026 | -1.4316 | 7.1034 |
| y20 | 587 | 584 | 597.36 | 594.071 | 610.8496 | 598.3905 | 598.3959 | 583.2983 | 602.5850 | 580.6381 | 588.9739 |
| x21 | 99 | 89 | 95.312 | 90.9369 | 98.3239 | 97.8696 | 97.8656 | 80.4288 | 82.0523 | 90.4552 | 92.5254 |
| y21 | 631 | 637 | 625.99 | 621.3943 | 624.958 | 624.055 | 624.0605 | 623.6525 | 656.6469 | 609.6711 | 620.7568 |
| Best weight (lb) | 1925.79 | 1871.7 | 1871.7 | 1975.84 | 1908.83 | 1864.1 | 1861.429 | 2,297.8011 | 2,283.0965 | 1,998.2013 | 1,911.7774 |
| Average weight (lb) | N/A | N/A | 1884.16 | N/A | N/A | 1894.056 | 1873.011 | 2,693.6165 | 2,736.9421 | 2,098.9179 | 2,070.6756 |
| Worst weight (lb) | N/A | N/A | 1899.7 | N/A | N/A | 2007.563 | 1908.991 | 3,120.5369 | 3,281.0248 | 2,321.7244 | 2,425.9799 |
| Std. deviation (lb) | N/A | N/A | 7.77 | N/A | N/A | 34.755 | 20.782 | 222.85 | 243.15 | 83.58 | 110.69 |
| No. of analyses | 100,000 | N/A | 187,488 | 25,000 | 25,000 | 25,000 | 20,040 | 30,000 | 29,840 | 29,600 | 30,000 |

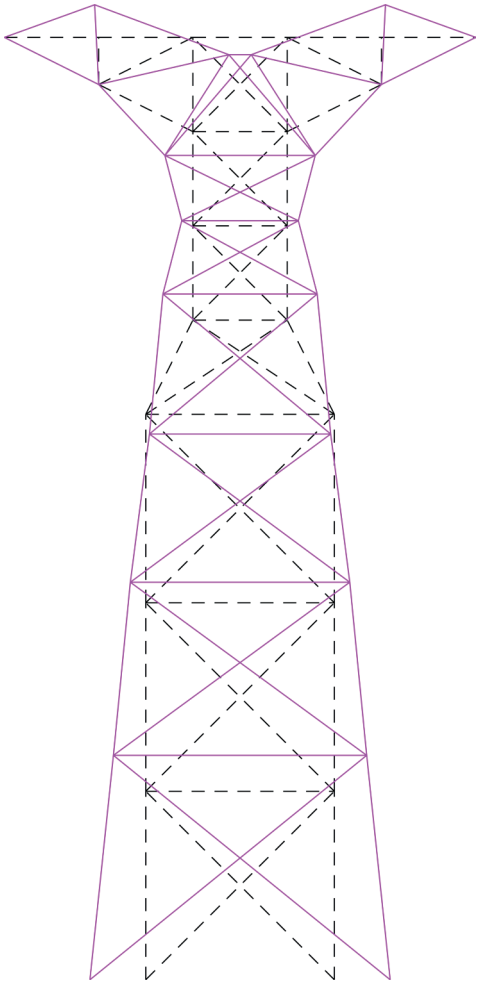

**Fig. 15** Optimum layout of the 47-bar planar truss

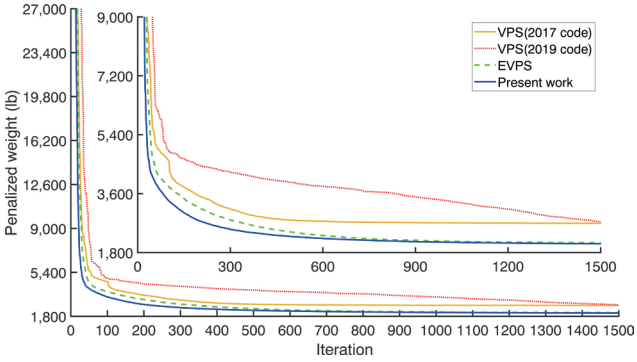


**Fig. 16** Convergence curves for the 47-bar planar truss

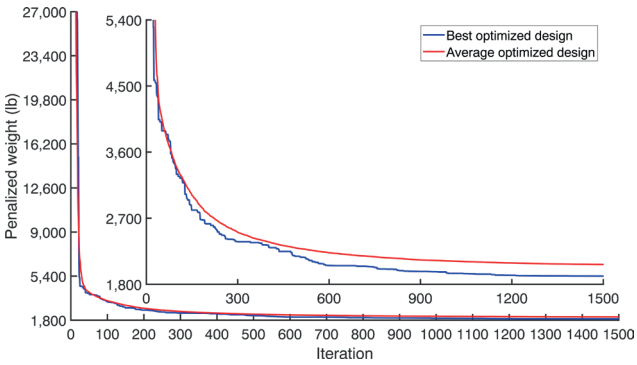


**Fig. 17** Convergence curves of the best and average performance of the studied algorithm for the 47-bar planar truss

## References

[1] Kaveh, A. "Advances in Metaheuristic Algorithms for Optimal Design of Structures", 3rd ed., Springer, Cham, Switzerland, 2021. https://doi.org/10.1007/978-3-030-59392-6

[2] Kennedy, J., Eberhart, R. "Particle swarm optimization", In: Proceedings of ICNN'95 - International Conference on Neural Networks, Perth, WA, Australia, 1995, pp. 1942–1948. https://doi.org/10.1109/ICNN.1995.488968

[3] Holland, J. H. "Genetic Algorithms", Scientific American, 267(1), pp. 66–73, 1992. [online] Available at: http://www.jstor.org/stable/24939139

[4] Kaveh, A., Mahdavi, V. R. "Colliding bodies optimization: A novel meta-heuristic method", Computers & Structures, 139, pp. 18–27, 2014. https://doi.org/10.1016/j.compstruc.2014.04.005

[5] Dorigo, M., Maniezzo, V., Colorni, A. "Ant system: optimization by a colony of cooperating agents", IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 26(1), pp. 29–41, 1996. https://doi.org/10.1109/3477.484436

[6] Kaveh, A., Talatahari, S. "A novel heuristic optimization method: charged system search", Acta Mechanica, 213(3), pp. 267–289, 2010. https://doi.org/10.1007/s00707-009-0270-4

[7] Woo, Z. G., Hoon, J. K., Loganathan, G. V. "A New Heuristic Optimization Algorithm: Harmony Search", Simulation, 76(2), pp. 60–68, 2001. https://doi.org/10.1177/003754970107600201

[8] Černý, V. "Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm", Journal of Optimization Theory and Applications, 45(1), pp. 41–51, 1985. https://doi.org/10.1007/BF00940812

[9] Kaveh, A., Khayatazad, M. "A new meta-heuristic method: Ray Optimization", Computers & Structures, 112–113, pp. 283–294, 2012. https://doi.org/10.1016/j.compstruc.2012.09.003

[10] Erol, O. K., Eksin, I. "A new optimization method: Big Bang–Big Crunch", Advances in Engineering Software, 37(2), pp. 106–111, 2006. https://doi.org/10.1016/j.advengsoft.2005.04.005

[11] Kaveh, A., Ilchi Ghazaan, M. "A new meta-heuristic algorithm: vibrating particles system", Scientia Iranica, 24(2), pp. 551–566, 2017. https://doi.org/10.24200/sci.2017.2417

[12] Kaveh, A., Ilchi Ghazaan, M. "Meta-heuristic Algorithms for Optimal Design of Real-Size Structures", Springer, Cham, Switzerland, 2018.
https://doi.org/10.1007/978-3-319-78780-0

[13] Kaveh, A., Zaerreza, A. "Size/layout optimization of truss structures using shuffled shepherd optimization method", Periodica Polytechnica Civil Engineering, 64(2), pp. 408–421, 2020.
https://doi.org/10.3311/PPci.15726

[14] Panagant, N., Bureerat, S. "ADOSH: software with graphic user interface for analysis and design of truss structures", Asian Journal of Civil Engineering, 19(3), pp. 273–286, 2018.
https://doi.org/10.1007/s42107-018-0024-5

[15] Wu, S.-J., Chow, P.-T. "Integrated discrete and configuration optimization of trusses using genetic algorithms", Computers & Structures, 55(4), pp. 695–702, 1995.
https://doi.org/10.1016/0045-7949(94)00426-4

[16] Hasançebi, O., Erbatur, F. "Layout optimization of trusses using improved GA methodologies", Acta Mechanica, 146(1–2), pp. 87–107, 2001.
https://doi.org/10.1007/BF01178797

[17] Hasançebi, O., Erbatur, F. "On efficient use of simulated annealing in complex structural optimization problems", Acta Mechanica, 157(1–4), pp. 27–50, 2002.
https://doi.org/10.1007/BF01182153

[18] Kaveh, A., Kalatjari, V. "Size/geometry optimization of trusses by the force method and genetic algorithm", ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik, 84(5), pp. 347–357, 2004.
https://doi.org/10.1002/zamm.200310106

[19] Tang, W., Tong, L., Gu, Y. "Improved genetic algorithm for design optimization of truss structures with sizing, shape and topology variables", International Journal for Numerical Methods in Engineering, 62(13), pp. 1737–1762, 2005.
https://doi.org/10.1002/nme.1244

[20] Rahami, H., Kaveh, A., Gholipour, Y. "Sizing, geometry and topology optimization of trusses via force method and genetic algorithm", Engineering Structures, 30(9), pp. 2360–2369, 2008.
https://doi.org/10.1016/j.engstruct.2008.01.012

[21] Kazemzadeh Azad, S., Kazemzadeh Azad, S., Jayant Kulkarni, A. "Structural Optimization Using a Mutation-Based Genetic Algorithm", International Journal of Optimization in Civil Engineering, 2(1), pp. 81–101, 2012. [online] Available at: http://ijoce.iust.ac.ir/article-1-80-en.html

[22] Miguel, L. F. F., Lopez, R. H., Miguel, L. F. F. "Multimodal size, shape, and topology optimisation of truss structures using the Firefly algorithm", Advances in Engineering Software, 56, pp. 23–37, 2013.
https://doi.org/10.1016/j.advengsoft.2012.11.006

[23] Gholizadeh, S. "Layout optimization of truss structures by hybridizing cellular automata and particle swarm optimization", Computers & Structures, 125, pp. 86–99, 2013.
https://doi.org/10.1016/j.compstruc.2013.04.024

[24] Mortazavi, A., Toğan, V., Nuhoğlu, A. "Weight minimization of truss structures with sizing and layout variables using integrated particle swarm optimizer", Journal of Civil Engineering and Management, 23(8), pp. 985–1001, 2017.
https://doi.org/10.3846/13923730.2017.1348982

[25] Panagant, N., Bureerat, S. "Truss topology, shape and sizing optimization by fully stressed design based on hybrid grey wolf optimization and adaptive differential evolution", Engineering Optimization, 50(10), pp. 1645–1661, 2018.
https://doi.org/10.1080/0305215X.2017.1417400

[26] Jawad, F. K. J., Ozturk, C., Dansheng, W., Mahmood, M., Al-Azzawi, O., Al-Jemely, A. "Sizing and layout optimization of truss structures with artificial bee colony algorithm", Structures, 30, pp. 546–559, 2021.
https://doi.org/10.1016/j.istruc.2021.01.016

[27] Kaveh, A., Vaez, S. R. H., Hosseini, P., Ezzati, E. "Layout optimization of planar braced frames using modified dolphin monitoring operator", Periodica Polytechnica Civil Engineering, 62(3), pp. 717–731, 2018.
https://doi.org/10.3311/PPci.11654

[28] Kaveh, A., Hoseini Vaez, S., Hosseini, P. "Enhanced vibrating particles system algorithm for damage identification of truss structures", Scientia Iranica, 26(1), pp. 246–256, 2019.
https://doi.org/10.24200/SCI.2017.4265

[29] Kaveh, A., Talatahari, S. "Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures", Computers & Structures, 87(5), pp. 267–283, 2009.
https://doi.org/10.1016/j.compstruc.2009.01.003

[30] Kaveh, A., Hoseini Vaez, S. R., Hosseini, P. "Matlab code for an enhanced vibrating particles system algorithm", International Journal of Optimization in Civil Engineering, 8(3), pp. 401–414, 2018. [online] Available at: http://ijoce.iust.ac.ir/article-1-352-en.html

[31] Kaveh, A., Ilchi Ghazaan, M. "MATLAB Code for vibrating particles system algorithm", International Journal of Optimization in Civil Engineering, 7(3), pp. 355–366, 2017. [online] Available at: http://ijoce.iust.ac.ir/article-1-302-en.html

[32] Kaveh, A., Bakhshpoori, T. "Metaheuristics: Outlines, MATLAB Codes and Examples", Springer, Cham, Switzerland, 2019.
https://doi.org/10.1007/978-3-030-04067-3