Engineering with Computers
https://doi.org/10.1007/s00366-022-01760-0

ORIGINAL ARTICLE

Physics-informed neural energy-force network: a unified solver-free numerical simulation for structural optimization

Hau T. Mai[1,2] · Dai D. Mai[3] · Joowon Kang[4] · Jaewook Lee[1] · Jaehong Lee[1]

Received: 27 July 2022 / Accepted: 28 October 2022

Abstract
In this study, a physics-informed neural energy-force network (PINEFN) framework is first proposed to directly solve the optimum design of truss structures that structural analysis is completely removed from the implementation of the global optimization. Herein, a loss function is constructed to guide the training network based on the complementary energy, constitutive equations, and weight of the structure. Now only neural network (NN) is used in our scheme to minimize the loss function wherein weights and biases of the network are considered as design variables. In this model, spatial coordinates of truss members are examined as input data, while corresponding cross-sectional areas and redundant forces unknown to the network are taken account of output. Accordingly, the predicted outputs obtained by feedforward are employed to establish the loss function relied on physics laws. And then, back-propagation and optimizer are applied to automatically calculate sensitivity and adjust parameters of the network, respectively. This whole process, which is the so-called training, is repeated until convergence. The optimum weight of the structure corresponding to the minimum loss function is indicated as soon as the training process ends without using any structural analyses. Several benchmark examples for sizing optimization of truss structures are examined to determine the reliability, efficiency, and applicability of the proposed model. Obtained outcomes indicated that it not only reduces the computational cost dramatically but also yields higher accuracy and faster convergence speed compared with recent literature.

Keywords Physics-informed · Force method · Deep neural networks · Machine learning · Unsupervised learning · Complementary energy · Truss optimization

1 Introduction

Over the past decade, the design optimization of truss structures has received considerable attention from many researchers in the computational mechanics community. Its objective is to minimize the structural weight while satisfying all constraints. In general, although there have been a variety of algorithms employed to address this issue, they all work on the same basic principle, as shown in Fig. 1a. Therein, the optimization tool often requires numerical simulations, such as finite element analysis (FEA), to estimate structural responses during each iteration of the optimizer. And they can be divided into two main classes. In the first one, the gradient-based algorithms have been successfully applied for searching optimal solutions. For instance, an algorithm based on the optimality criterion (OC) is developed by Khot et al. [1–3] and Rizzi [4] to find the optimal weight of the truss structure. Hrinda et al. [5] has proposed a new algorithm by combining the design-variable update scheme and arc-length method. Besides, a coupling methodology based on the OC and nonlinear analysis technique was delivered by Saka and Ulker [6] to save the computational cost. Schmit and Farshi [7] developed a sequence of linear programs to sizing structural systems. However, this approach cannot deal with the lack of gradient information from the objective and constraint functions. The other one is the gradient-free algorithms which rely on evolutionary and population genetics to address the optimal design of truss structures, such as firefly algorithm [8, 9], harmony search [10, 11], genetic algorithms [12, 13], particle swarm algorithm [14–18], chaotic coyote algorithm [19], big bang-big crunch [20, 21], teaching-learning-based optimization [22], hybrid differential evolution and symbiotic organisms search [23], adaptive hybrid evolutionary firefly algorithm [8],

✉ Jaehong Lee
jhlee@sejong.ac.kr

Extended author information available on the last page of the article

Published online: 27 January 2023

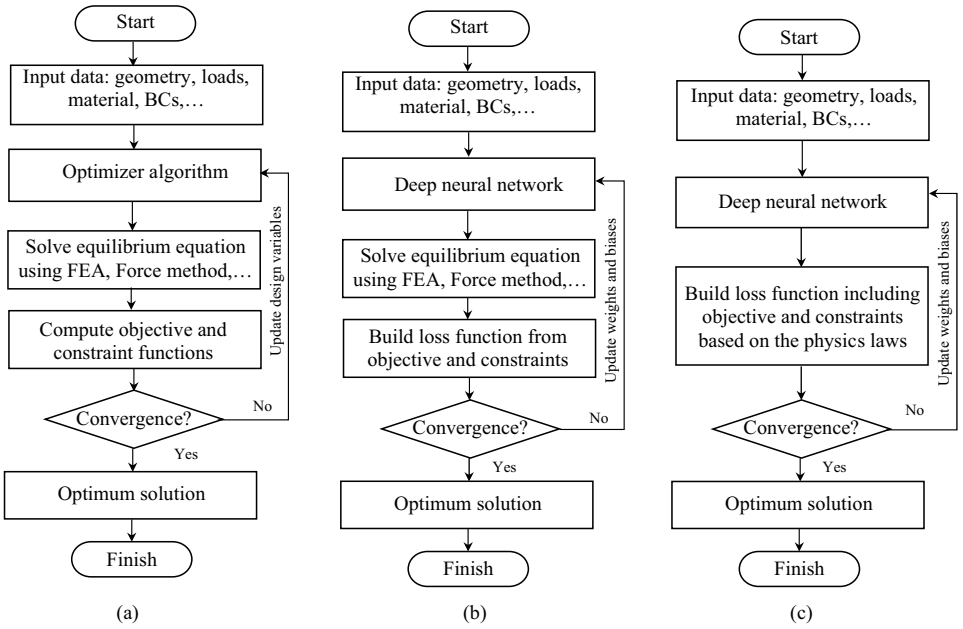

Fig. 1 Process of structural optimization. **a** Conventional approach including optimizer algorithm and structural analysis. **b** Framework combines between the deep neural network and structural analysis. **c** Physics-informed neural network without using any structural analyses

evolutionary symbiotic organisms search [24], and so on. Despite these algorithms have achieved certain success, they require many evaluation functions, slow convergence rate, and high computational cost due to large-scale problems.

In recent years, machine learning (ML) has been demonstrated to be a powerful tool for making decisions with applications in a variety of areas, including medical diagnoses, voice recognition, language processing, industrial automation, etc. Therein, deep neural network (DNN) in the form of ML models has received much attention in computational mechanics fields, such as structural analysis [25], structural health monitoring [26], reliability analysis [27], and structural optimization [28] is no exception. As far as our knowledge goes, the applications of the NN to the structural optimization problems can be classified into two baselines. The first one is an approach based on data-driven models. Accordingly, the networks are commonly used in the idea of supervised learning to learn the mapping from arbitrary designs obtained by numerical simulations and act as a surrogate model of the solver. In fact, this methodology is not a new idea and has been developed since the 1990s for optimization of truss structure. For example, NN models were derived through supervised training for solving the optimal design of structural truss by Hajela and Berke [29, 30]. Adeli and Park [31] developed a nonlinear neural dynamics model to minimum mass design of space trusses. In addition, a combination between the NN and genetic algorithm for the optimization of industrial roofs was released by Ramasamy [32]. In recent times, an integrated model of differential evolution algorithm and DNN to optimize truss structures with geometric nonlinearity was introduced by Mai et al. [33]. Besides, it has also been successfully applied to tackle topology optimization problems of structures. Of particular interest is the study by Li et al. [34], in which the authors proposed a non-iterative topology optimization framework using the NN for heat conduction structure design. White [35] and Chi [36] used the NN as a surrogate model to replace the structural analysis phase for the topology optimization. However, in order to achieve the optimal solution, it may require a huge amount of data to train as well as the high computational cost to generate the input data and their corresponding supervised labels via the structural analysis. Further, a variety of questions arise as to how to determine the sufficient data size for different problems, how to ensure the generalization ability of the trained model, and is the obtained results really better than the conventional alternatives? And it is very difficult to answer these questions definitively. In addition, this approach still requires the structural analysis model, and the physics information is removed in the training process. The second baseline methodology is a combination between the NN and numerical solver to perform structural optimization directly. Figure 1b provides an overall view of this approach. Herein, the network is employed as an optimizer, while its loss function is built by the output values of the network and the structural responses obtained from the structural analysis, respectively. And several researchers have successfully applied this model with promising potential for the structural optimization. For example, Chandrasekhar et al. [37–39] firstly introduced a direct topology optimization framework using the NN. In addition, a deep unsupervised learning is proposed by Mai et al. [28] to perform the optimization of truss structures. Although this paradigm has shown the effectiveness of the NN for structural design optimization problems, it

still demands the structural analysis. The flowchart of this procedure is illustrated in Fig. 1b. It is easily be seen that most steps are similar as those involved in the conventional method (Fig. 1a). In fact, there is little difference which the network plays a crucial role in the implementation process as an optimizer. Hence, it also encounters difficulties due to the structural analysis.

Physics-informed neural networks (PINNs) are introduced as new, powerful, and efficient method for numerical simulations. In contrast to the data-driven approach, this technique allows to solve the problems by minimizing the loss function which is constructed based on the physics laws. In comparison to conventional numerical solvers, it can easily handle the problems with irregular domains as well as completely avoids a discretization like FEA [40]. Furthermore, the training data are easily collected from the known design information of the structure without any structural analyses, for example, boundary conditions (BCs), geometry, properties of materials, etc. In addition, one of its outstanding characteristic is that the sensitivity can be quickly and easily calculated with a back-propagation (BP) algorithm of the network. Hence, it has received much attention in the engineering field. PINN was first introduced by Raissi et al. [41] to solve partial differential equations (PDEs) with the associated boundary and initial conditions into the loss function. Nguyen-Thanh et al. introduced a deep energy method for the solutions of nonlinear finite deformation hyperelasticity problems. The same idea was adopted by Mai et al. [25] to perform the geometrically nonlinear analysis of inelastic truss structures. Indeed, PINN has recently been proven effective for solving more complex computational problems, including PDEs, structural analysis, fluid mechanics, and so on. However, it has still not been yet utilized for the structural optimization design thus far.

Motivated by the foregoing challenges and based on existing works, this paper aims at presenting a physics-informed neural energy-force network framework for solving size optimization of truss structures without using structural analysis. Instead of tackling the structural optimization problems as traditional approaches which are the combination between optimization algorithm and structural analysis, here only NN is built in our scheme to find the optimum weight of the structure, as illustrated in Fig. 1c. Accordingly, now the weights and biases which are known as trainable parameters of the network are regarded as design variables in place of the cross-sectional areas of truss members. Therein, the spatial coordinates of all truss members are treated as the input data which are easily gathered from the structure's connectivity information. The important thing that must be highlighted here is that this work relies on unsupervised learning, so the NN only has input data and does not require corresponding output values. In the proposed framework, the output of the network are the unknown cross-sectional areas and redundant forces which are represented by the parameters of the network. Accordingly, the training process aims to seek the optimal parameters of the network such that the loss function is minimized. At first, the spatial coordinates of truss elements are assigned to input, and the predicted outputs of the network are calculated by the feedforward (FF) algorithm. Thereafter, our loss function is designed based on these predicted values and physics laws to guide the training. Next, the BP algorithm of the network allowed to automatically calculate the sensitivity of the cost function with respect to the parameters. Subsequently, the optimizer relies on gradient estimates to adjust the weights and biases of the network. The above all steps are repeated until

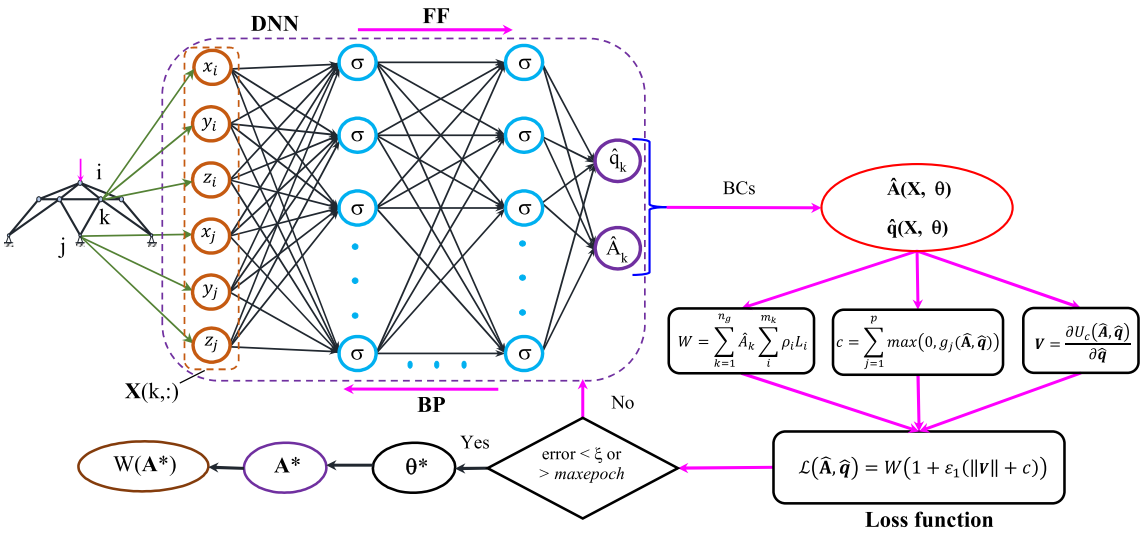

Fig. 2 Physics-informed neural energy-force networks framework for design optimization

the minimum loss is achieved, and it is called the learning process. And finally, the optimum weight of the structure corresponding to the NN's optimal parameters is found as soon as the training process ends without using any structural analyses. The reliability, efficiency, and applicability of the proposed model are also demonstrated through several examples for the truss design under various constraints. The obtained results showed that PINEFN not only saves a large computational cost but also yields higher accuracy and faster convergence speed. In addition, it does not use any structural analysis tools and does not need to prepare the data in advance.

The rest of this study is structured as follows. Section 2 provides the theoretical formulation of the structural optimization based on energy-force methods. Next, a physics-informed neural energy-force network approach is suggested in Sect. 3. In Sect. 4, several numerical examples are examined to demonstrate the efficiency of the proposed method. Finally, crucial conclusions are outlined in Sect. 5.

2 Structural optimization based on energy-force methods

Optimization of structures is known as a complex task related to the obtained structural response from numerical simulations. And sizing optimization of truss structures is one of these problems. Its main goal is to minimize the structural weight while satisfying all design constraints as well as the equilibrium equations simultaneously. Therein, the structural member's cross-sectional areas are treated as continuous design variables and confined within an acceptable range.

The general mathematical model for the linear elastic structure based on principle of minimum potential energy can be expressed as follows

$$\begin{aligned} \text{Minimize} \quad & W(\mathbf{A}) = \sum_{k=1}^{n_g} A_k \sum_{i=1}^{m_k} \rho_i L_i, \\ \text{subjected to} \quad & \frac{\partial \Pi_p}{\partial \mathbf{u}}(\mathbf{A}) = \mathbf{K}\mathbf{u} - \mathbf{f} = \mathbf{0}, \\ & g_j(\mathbf{A}) \le 0, \qquad j = 1, 2, ..., n_c, \\ & A_k^{low} \le A_k \le A_k^{up}, \qquad k = 1, 2, ..., n_g, \end{aligned} \tag{1}$$

where $W(.)$ is the weight of the truss structure; A_k is the cross-sectional area of the truss elements belonging to the kth group which can range between A_k^{low} and A_k^{up}; n_g denotes the total number of groups in the structure; m_k and n_c are the number of members in the kth group and constraint functions; ρ_i and L_i are the material density and length of the ith member; Π_p is the total potential energy; $\mathbf{u}$ and $\mathbf{f}$ are the vector of displacements and external forces at nodes, respectively; $\mathbf{K}$ denotes the stiffness matrix of structure; g_j represents the jth constraint function including displacement and stress. To obtain the constraints in Eq. (1), the important thing that must be highlighted here is that the principle of minimum potential energy is utilized to determine the displacement field $\mathbf{u}$ concerning the equilibrium equations and related BCs. And once it is found, the other structural responses can be easily achieved by the constitutive equations. It should be noted that the first constraint related to the first derivative of Π_p to $\mathbf{u}$ is automatically satisfied and ignored. Accordingly, in most of the structural optimization problems in the existing literature, this approach is carried

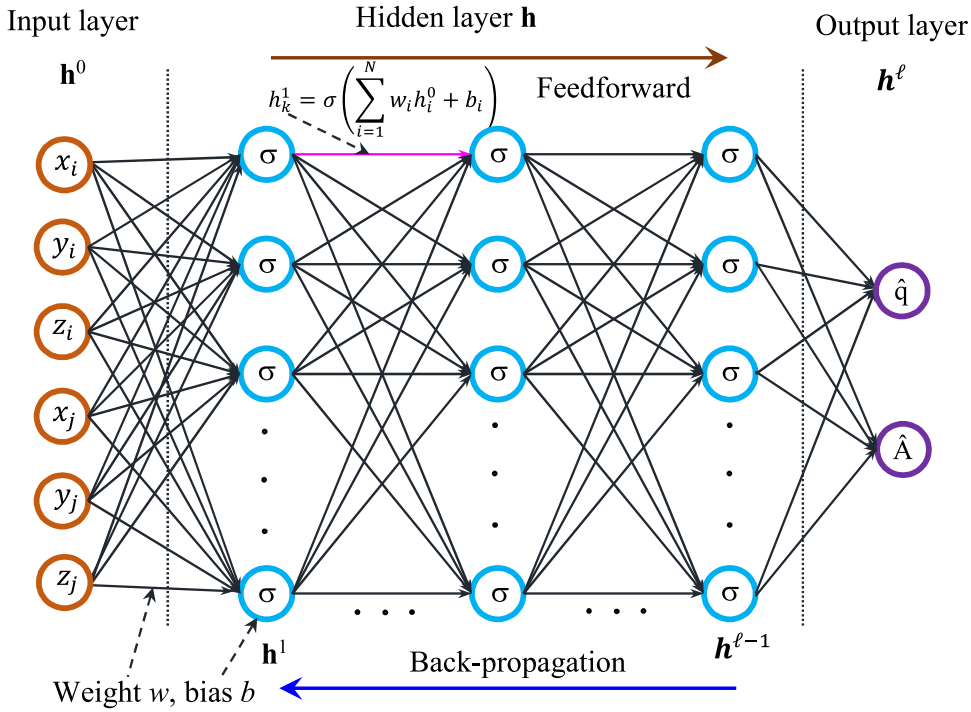

Fig. 3 A fully-connected deep neural network architecture

out by using the conventional FEA and incorporated into an optimization algorithm.

As pointed out by Ohkubo et al. [42], it was obvious that all structural behaviors obtained by using the principle of minimum complementary energy were similar to the potential energy. According to this approach, the actual member forces of trusses are first determined from the minimum of complementary energy to satisfy all stated compatibility conditions. And then the displacement field, as well as other responses, are easily calculated based on the obtained reaction and member forces. Therefore, the size optimization of truss structures based on the principle of minimum complementary energy can be rewritten similarly to Eq. (1). The derivative of Π_p to the displacement field is replaced by the derivative of Π_c to the redundant force $\mathbf{q}$. In order to Castigliano's second theorem, the complementary energy for internal forces is minimum and satisfies the compatibility conditions. Thus, its derivative concerning the unknown $\mathbf{q}$ is defined as follows

$$\frac{\partial \Pi_c}{\partial \mathbf{q}}(\mathbf{A}) = \left(\mathbf{B}_1^T \mathbf{G} \mathbf{B}_0\right)\bar{\mathbf{p}} + \left(\mathbf{B}_1^T \mathbf{G} \mathbf{B}_1\right)\mathbf{q} = \mathbf{0}. \tag{2}$$

in which $\bar{\mathbf{p}}\left(\in \mathbb{R}^{dn_s}\right)$ is the external load vector; $\mathbf{q}(\in \mathbb{R}^r)$ denotes the redundant force vector; $\mathbf{G}(\in \mathbb{R}^{m_s \times m_s})$ refers to the diagonal flexibility matrix; $\mathbf{B_0}\left(\in \mathbb{R}^{m_s \times dn_s}\right)$ and $\mathbf{B_1}(\in \mathbb{R}^{m_s \times r})$ are the generalized inverse and null matrices; r, m_s, and n_s are the number of total indeterminacies, the total number of members and boundary conditions, and total number of nodes of the d-dimensional truss structure, respectively. For more detail detailed derivation of Eq. (2) based on the force method, interested readers can consult Ref. [43]. And likewise to the previous method when the structural analysis is available, given the fact that this constraint is always satisfied with equality for the minimum complementary energy and can be ignored. Once the redundant force $\mathbf{q}$ is determined, the structural responses can be easily obtained from kinematic relations.

Note that there is a small difference between the two above approaches for the computational structural analysis. In particular, the first one applied the FEA with displacement as the primary unknown to achieve the minimum potential energy. While the remaining one considered the redundant force as the primary unknown of the force method which is found from the minimum complementary energy. However, the above-mentioned approaches are all based on the basic principle which required both the optimization algorithm and the numerical solver. Therein, the structural analysis is required for each iteration of the optimization process. Despite its success in a wide range of optimization problems, it still faces challenging issues due to the increasing computational cost for the high-dimensional problem.

To overcome the computing challenge, an alternative paradigm based on the complementary energy and force is developed for optimization of truss structures without using any structural analyses. Its mathematical formulation can be expressed as [44]

$$\begin{aligned}
\text{Minimize} \quad & W(\mathbf{A}) = \sum_{k=1}^{n_g} A_k \sum_{i=1}^{m_k} \rho_i L_i, \\
\text{subjected to} \quad & \frac{\partial \Pi_c}{\partial \mathbf{q}}(\mathbf{A}, \mathbf{q}) = \left(\mathbf{B}_1^T \mathbf{G} \mathbf{B}_0\right)\bar{\mathbf{p}} + \left(\mathbf{B}_1^T \mathbf{G} \mathbf{B}_1\right)\mathbf{q} = 0, \\
& g_j(\mathbf{A}, \mathbf{q}) \le 0, \qquad j = 1, 2, ..., n_c, \\
& A_k^{low} \le A_k \le A_k^{up}, \qquad k = 1, 2, ..., n_g, \\
& q_l^{low} \le q_l \le q^{up}, \qquad l = 1, 2, ..., r.
\end{aligned} \tag{3}$$

It is worth mentioning that this approach is quite different comparing to the previous traditional methods for searching the optimum weight of truss structures. More concretely, there are three main differences between Eq. (3) and Eq. (1) as below

(i) First, it is easily seen that the design variables of Eq. (3) include the cross-sectional areas and redundant forces, while the other methods are only the cross-sectional area of the truss members as shown in Eq. (1).
(ii) Next, the first constraint automatically satisfies a zero first derivative for the potential or complementary energies in Eq. (1) where the structural analysis is carried out to estimate the structural responses. Hence, it can be entirely ignored. On the contrary, this characteristic does not exist in Eq. (3) due to considering the redundant forces as the design variables.
(iii) Finally, the optimization algorithm and structural analysis are required to achieve the optimal solution for Eq. (1). And all the while, the approach based on Eq. (3) only needs to be the optimization algorithm.

In order to tackle Eq. (3), a self-adaptive penalty function is used to handle the constraint function through transforming the constrained structural optimization problem to an unconstrained optimization one [45, 46]. Consequently, it is rewritten as follows

$$\begin{aligned}
\text{Minimize} \quad & \mathcal{L}(\mathbf{A}, \mathbf{q}) = \left[1 + \varepsilon_1(\|\mathbf{v}\| + c)\right]^{\varepsilon_2} W(\mathbf{A}), \\
& \mathbf{v}(\mathbf{A}, \mathbf{q}) = \left(\mathbf{B}_1^T \mathbf{G} \mathbf{B}_0\right)\bar{\mathbf{p}} + \left(\mathbf{B}_1^T \mathbf{G} \mathbf{B}_1\right)\mathbf{q}, \\
& c = \sum_{j=1}^{n_c} \max\left(0, g_j(\mathbf{A}, \mathbf{q})\right),
\end{aligned} \tag{4}$$

in which c denotes the sum of the violated constraints; ε_1 and ε_2 represent parameters to control the exploration and exploitation rates of the design region. As suggested by Sonmez [45] and Hasancebi [46], the parameter ε_2 is set equal to 1, and the value of ε_1 is self-adaptive based on the feedback

information from the previous iteration and is expressed as follows

$$\varepsilon_1^{(t)} = \begin{cases} (1/\kappa)\varepsilon_1^{(t-1)} & \text{if } \mathcal{L}^{(t-1)} \text{ feasible,} \\ \kappa\,\varepsilon_1^{(t-1)} & \text{if } \mathcal{L}^{(t-1)} \text{ infeasible,} \end{cases} \tag{5}$$

where $\varepsilon_1^{(t)}$ is the penalty coefficient at the tth iteration, and $\varepsilon_1^{(1)}$ is chosen 1 at the beginning of the iteration. κ denotes the learning parameter of $\varepsilon_1^{(t)}$, and is given by

$$\kappa = 1 + \frac{1}{n_c + 1} > 1.01. \tag{6}$$

3 Physics-informed neural energy-force network

In this section, the physics-informed neural energy-force network is developed to perform optimization of truss structures. The whole schematic illustration, as shown in Fig. 2, gives an overall view of the present approach. Different from most existing works, the NN model with unsupervised learning is designed to find the optimum weight of structures that it only relies on the knowledge of any physical laws and doesn't need any structural analysis. Note that for the present approach, the design variables of the structural optimization are the parameters of the network $\boldsymbol{\theta}$ including weights and biases, instead of the cross-sectional areas of truss members. According to the flow diagram, a fully connected NN is set first up to perform the design optimization of truss structures. At the start of the training run, all weights and all biases were initialized with their values obey the normal distribution in the range of $[-1,\ 1]$. The coordinates of all truss members in the structure are employed as the input training data, whilst the corresponding cross-sectional areas and redundant forces are defined as the outputs of the network $(\hat{\mathbf{A}}, \hat{\mathbf{q}})$ obtained by the FF process. Based on these outputs, external forces, BCs, and physical laws, the self-adaptive penalty function, including the objective, complementary energy, and constraints, is derived as the loss function which is minimized by tuning the network's weights and biases. And then BP algorithm of the network allows to automatically calculate the sensitivity of the loss function to the corresponding parameters, whereby they will be updated. One epoch is an iteration of training that consists of pairs of FF and BP. In order to carry out the training process, the above operations are performed and repeated many times until reaches the minimum loss function value corresponding to the optimum weight of truss structures. In general, the proposed approach consists of three main components, namely training data, DNN, and loss function. The following sub-sections and Algorithm 1 allow to represent in greater detail the proposed methodology.

Table 1 Hyperparameters fCVEor the benchmarks tested in this study

Problem	Depth	Width	Optimizer	Learning rate	Epochs
10-bars	2	30	Adam	0.01	1000
17-bars	2	40	Adam	0.01	1000
200-bars	3	40	Adam	0.01	5000
25-bars	3	30	Adam	0.01	1000
72-bars	3	60	Adam	0.01	1000
120-bars dome	3	60	Adam	0.01	1000

3.1 Training data

It is important to stress that, contrary to previous approaches which utilized supervised learning framework, the model presented here is designed based on the unsupervised learning algorithm. Hence, the training data only has the input data, while the corresponding output values are not given. In other words, the cross-sectional areas, redundant forces, and the responses of the structure including stress, strain, displacement, member force, etc. are not included in the training data. It means that all training data of PINEFN only contains a set of coordinates of all truss members. In particular, the coordinates of the kth member that connects the nodes ith and jth $\mathbf{X}(k,:) = \left[x_i, y_i, z_i, x_j, y_j, z_j\right] \in \mathbb{R}^6$ are treated as a training data point which is used as the input values of the NN. Obviously, the whole training data $\mathbf{X}$ can be easily collected from the geometric and connection information of nodes. Furthermore, its size is small $\mathbf{X}\left(\in \mathbb{R}^{m_s \times 4}\right)$ and $\mathbf{X}\left(\in \mathbb{R}^{m_s \times 6}\right)$ for planar and space truss structures, respectively. Here m_s denotes the total number of truss members, while 4 or 6 show the number of spatial coordinates for 2- or 3- dimensional truss. Due to the nature of the activation function, the input data is normalized to the interval of

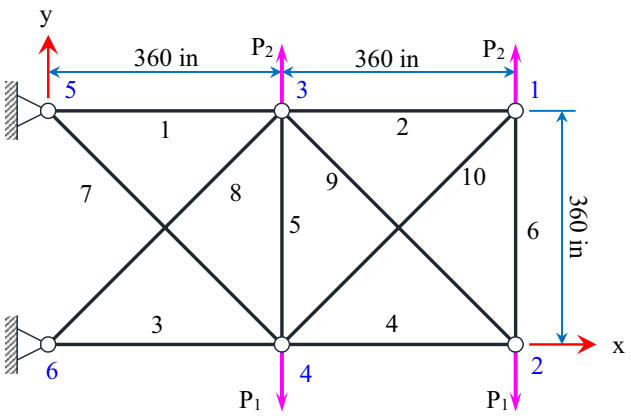

Fig. 4 A 10-bar planar truss structure

Table 2 Comparison of the obtained results for the 10-bar truss with the first loading condition

A_i (in^2)	Schmit [7]	Rizzi [4]	Lee [10]	Li [14]			This study	
				PSO	PSOPC	HPSO	DE	PINEFN
A_1	33.43	30.73	30.15	33.469	30.569	30.704	30.529	30.534
A_2	0.1	0.1	0.102	0.11	0.1	0.1	0.100	0.101
A_3	24.26	23.93	22.71	23.177	22.974	23.167	23.216	23.208
A_4	14.26	14.73	15.27	15.475	15.148	15.183	15.200	15.230
A_5	0.1	0.1	0.102	3.649	0.1	0.1	0.100	0.100
A_6	0.1	0.1	0.544	0.116	0.547	0.551	0.555	0.551
A_7	8.388	8.542	7.541	8.328	7.493	7.46	7.455	7.458
A_8	20.74	20.95	21.56	23.34	21.159	20.978	21.029	21.053
A_9	19.69	21.84	21.45	23.014	21.556	21.508	21.535	21.495
A_{10}	0.1	0.1	0.1	0.19	0.1	0.1	0.100	0.100
W_{best} (lb)	5089.00	5076.66	5057.88	5529.50	5061.00	5060.92	5060.86	5061.06
CVE_{max} (%)	21.14	None	0.09	None	None	None	None	None
Times (s)	–	–	–	–	–	–	96.52	5.02

[-1, 1] to accelerate the learning process. Additionally, the information to describe the problem, such as material, BCs, external loads, number of total indeterminacies, and so on, not included in the training data, but they were necessary to put some restrictions on the output values as well as building the loss function of the network. Obviously, they can be easily obtained before starting the training.

3.2 Feedforward

Firstly, a fully connected neural network with depth ℓ, as shown in Fig. 3, is designed to find the optimal truss structure. It consists of one input layer, one output layer, and $(\ell - 1)$ hidden layers. Therein, the first layer is known as the input layer which has four or six neurons corresponding with the coordinates of members for 2- or 3-dimensional truss, respectively. The final layer is known as the output layer with two neurons, which corresponds to the predicted cross-sectional area and redundant force. And the other layers are referred to as the hidden layers. Meanwhile, the number of hidden layer and hidden neurons depend on the complexity of the application. All neurons of the present layer are connected to all units in the previous layer via the parameters of the network, which consists of weights and biases, and their initial values $\boldsymbol{\theta}_0$ are randomly generated following the normal distribution in the range of [-1, 1]. In addition, Adam optimizer with its default parameters as suggested by Kingma and Ba [47] was set up to perform the training task.

In general, the FF is the process of propagation of the training data from the input layer to the hidden layers and then to the output layer of the NN. Accordingly, the relationship between the input and output values of each layer for the kth truss member obtained by FF is expressed as follows

Table 3 Error of the constraints for the 10-bar planar truss with the first loading condition

Elements	Stress (ksi)			Dofs	Displacement (in)		
	FEA	PINEFN	Error (%)		FEA	PINEFN	Error (%)
1	6.6366	6.6366	0.0002	u_1	0.1921	0.1921	0.0095
2	− 1.3002	− 1.2997	0.0398	v_1	− 2.0000	− 2.0000	0.0002
3	− 8.5040	− 8.5040	0.0004	u_2	− 0.5428	− 0.5428	0.0004
4	− 6.5744	− 6.5744	0.0003	v_2	− 1.9914	− 1.9914	0.0002
5	24.9995	24.9993	0.0011	u_3	0.2389	0.2389	0.0002
6	− 0.2385	− 0.2386	0.0058	v_3	− 0.7354	− 0.7354	0.0002
7	18.4610	18.4609	0.0008	u_4	− 0.3061	− 0.3061	0.0004
8	− 6.8950	− 6.8950	0.0002	v_4	− 1.6353	− 1.6353	0.0007
9	6.5879	6.5879	0.0000				
10	1.8553	1.8554	0.0068				
Max	24.9995	24.9993	0.0398	–	0.2389	0.2389	0.0095
Min	− 8.5040	− 8.5040	0.0000	–	− 2.0000	− 2.0000	0.0002

Table 4 Standard errors (%) of optimal weight for the 10-bar truss with respect to various optimizers and activation functions for the first load case

Activation functions	Optimizers						
	Adam	Adamw	Adadelta	Adagrad	Adamax	SGD	ASGD
Softmax	0.5781	0.2678	0.9666	5.5167	1.1592	12.0243	56.8574
Softplus	0.3242	0.3413	0.6172	0.8193	0.3717	0.9710	29.6784
Tanh	0.1425	0.1628	0.8713	1.3563	0.1969	2.3996	30.1295
Sigmoid	0.3545	0.3865	1.0425	1.3875	0.4594	15.0234	40.2913
ReLU	0.0043	0.3468	0.0221	0.0540	0.0212	0.6399	27.9569
LeakyReLU	0.0039	0.0052	0.0146	0.0421	0.0257	0.6212	27.7776

Table 5 Standard errors (%) of optimal weight for the 10-bar truss with respect to various hidden layers and neurons for the first load case

Hidden layers	Number of hidden neurons								
	10	15	20	25	30	35	40	45	50
1	0.4407	0.3476	0.3284	0.3202	0.3262	0.3187	0.3174	0.3180	0.3179
2	0.0047	0.0073	0.3142	0.0048	0.0039	0.3141	0.3141	0.2220	0.3132
3	0.0763	0.0274	0.0173	0.0099	0.0288	0.3438	0.1320	0.3132	0.0086

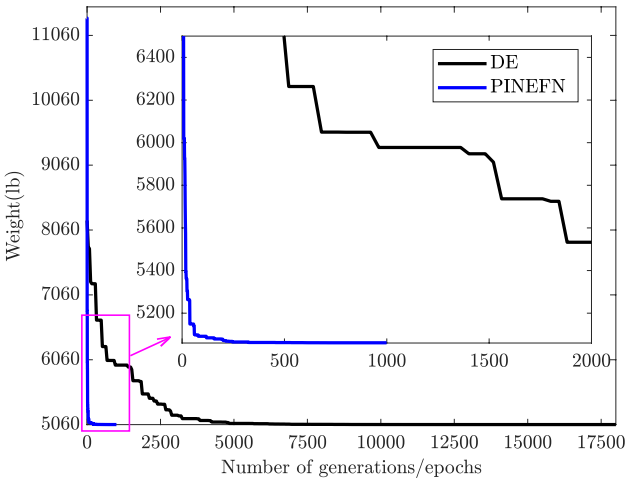

Fig. 5 The weight convergence histories of the 10-bar truss obtained using the PINEFN and DE for the first load case

$$\begin{aligned}
&\text{input layer} \quad : \quad \mathbf{h}^0 = \mathbf{X}(k,:) = \left[x_i, y_i, z_i, x_j, y_j, z_j\right] \in \mathbb{R}^6,\\
&\text{hidden layers:} \quad \mathbf{h}^n = f_1\left(\mathbf{W}^{n^T}\mathbf{h}^{(n-1)} + \mathbf{b}^n\right) \in \mathbb{R}^{m_n},\\
&\qquad\qquad\quad for \;\; 1 \le n \le (\ell - 1),\\
&\text{output layer} \;\; : \quad \mathbf{h}^\ell = f_2\left(\mathbf{W}^{\ell^T}\mathbf{h}^{(\ell-1)} + \mathbf{b}^\ell\right)\\
&\qquad\qquad\quad = [\hat{A}_k, \hat{q}_k], \; \in \mathbb{R}^2,
\end{aligned} \tag{7}$$

where $\mathbf{W}^{(.)}$ is the weight matrix; $\mathbf{b}^{(.)}$ is the bias vector; m_k is the number of units in the kth hidden layer; $f(.)$ denotes the activation function, which allows the network to learn about the complete relationship between the input and output. There are several popular activation functions, for example, Tanh, Sigmoid, ReLU, LeakyReLU, Softmax, Linear, and so on. In this study, the softmax function is selected for the output layer, while the activation function of the hidden layer is LeakyReLU.

From Eq. (7), it is clear that we can easily get the predicted output values including the cross-sectional areas $\hat{\mathbf{A}}(\in \mathbb{R}^{m_s})$ and redundant forces $\hat{\mathbf{q}}(\in \mathbb{R}^{m_s})$ with respect to the coordinates of other truss members in the training data $\mathbf{X}$. In fact, the total number of redundant fores r is less than the number of truss members m_s. Consequently, only first r values of the predicted redundant force vector $\hat{\mathbf{q}}$ are used to build the loss function, while all predicted cross-sectional areas are available. Obviously, this will not affect the performance of the NN, because only the values used to design the loss function allow tuning the parameters of the network.

3.3 Backpropagation & training

Next, the loss function which plays a role as an objective function of the network is formulated based on the output values from the FF phase. According to this approach, the predicted cross-sectional areas and redundant forces are expressed by the nodes' coordinates and the parameters of the network. And clearly, it based on the penalty function in Eq. (4) is rewritten as follows

$$\mathcal{L}(\mathbf{X}, \boldsymbol{\theta}) \;= \left(1 + \varepsilon_1(\|\mathbf{v}\| + c)\right)^{\varepsilon_2} W\left(\hat{\mathbf{A}}(\mathbf{X}, \boldsymbol{\theta})\right), \tag{8}$$

with

$$\begin{aligned}
\mathbf{v}(\mathbf{X}, \boldsymbol{\theta}) &= \left(\mathbf{B}_1^T\mathbf{G}\mathbf{B}_0\right)\bar{\mathbf{p}} + \left(\mathbf{B}_1^T\mathbf{G}\mathbf{B}_1\right)\hat{\mathbf{q}}(\mathbf{X}, \boldsymbol{\theta}),\\
c(\mathbf{X}, \boldsymbol{\theta}) &= \sum_{k=1}^{n} \max\left(0, g_j\left(\hat{\mathbf{A}}(\mathbf{X}, \boldsymbol{\theta}), \hat{\mathbf{q}}(\mathbf{X}, \boldsymbol{\theta})\right)\right).
\end{aligned} \tag{9}$$

As shown in Eq. (8), it is easily seen that the design variables are now the parameters of the network instead of the

Table 6 Comparison of the obtained results for 10-bar planar truss with the second loading condition

A_i (in^2)	Schmit [7]	Rizzi [4]	Lee [10]	Li [14]			This study	
				PSO	PSOPC	HPSO	DE	PINEFN
A_1	24.29	23.53	23.25	22.935	23.743	23.353	23.528	23.731
A_2	0.1	0.1	0.102	0.113	0.101	0.1	0.100	0.101
A_3	23.35	25.29	25.73	25.355	25.287	25.502	25.280	25.238
A_4	13.66	14.37	14.51	14.373	14.413	14.25	14.375	14.218
A_5	0.1	0.1	0.1	0.1	0.1	0.1	0.100	0.100
A_6	1.969	1.97	1.977	1.99	1.969	1.972	1.970	1.975
A_7	12.67	12.39	12.21	12.346	12.362	12.363	12.391	12.396
A_8	12.54	12.83	12.61	12.923	12.694	12.894	12.826	12.763
A_9	21.97	20.33	20.36	20.678	20.323	20.356	20.336	20.393
A_{10}	0.1	0.1	0.1	0.1	0.103	0.101	0.100	0.100
W_{best} (lb)	4691.84	4676.92	4668.81	4679.47	4677.70	4677.29	4676.93	4677.25
CVE_{max} (%)	None	None	0.19	None	None	None	None	None
Times (s)	–	–	–	–	–	–	104.29	5.83

Table 7 Error of the constraints for the 10-bar planar truss with the second loading condition

Elements	Stress (ksi)			Dofs	Displacement (in)		
	FEA	PINEFN	Error (%)		FEA	PINEFN	Error (%)
1	6.4582	6.4582	0.0002	u_1	− 0.0390	− 0.0387	0.7538
2	− 7.5425	− 7.5399	0.0349	v_1	− 1.1022	− 1.1023	0.0032
3	− 9.7764	− 9.7764	0.0002	u_2	− 0.6071	− 0.6071	0.0001
4	− 7.0867	− 7.0867	0.0000	v_2	− 2.0000	− 2.0000	0.0016
5	24.9969	24.9966	0.0011	u_3	0.2325	0.2325	0.0002
6	24.9406	24.9396	0.0041	v_3	− 0.6574	− 0.6574	0.0001
7	16.7409	16.7408	0.0009	u_4	− 0.3520	− 0.3520	0.0002
8	− 5.9016	− 5.9016	0.0001	v_4	− 1.5573	− 1.5573	0.0007
9	6.9875	6.9875	0.0000				
10	10.6667	10.6674	0.0062				
Max	24.9969	24.9966	0.0349	–	0.2325	0.2325	0.7538
Min	− 9.7764	− 9.7764	0.0000	–	− 2.0000	− 2.0000	0.0001

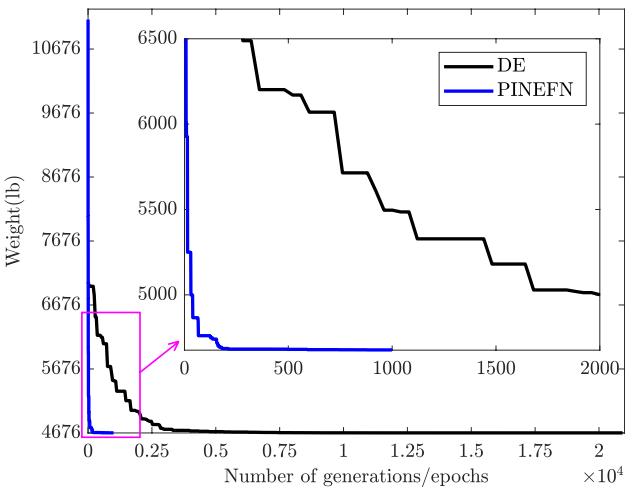

Fig. 6 The weight convergence histories of the 10-bar truss obtained using the PINEFN and DE for the second load case

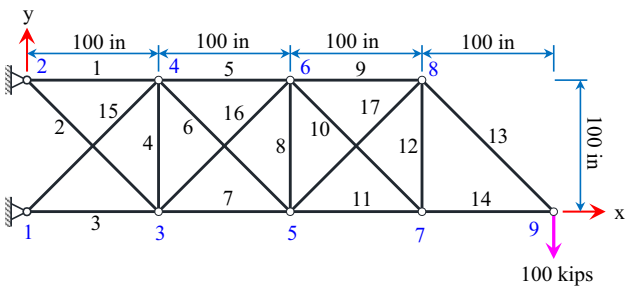

Fig. 7 A 17-bar planar truss structure

cross-sectional areas as the conventional approach. Furthermore, it is worth mentioning that this function is built relying on the physics laws and the information to define the problem. Hence, the structural analysis are completely

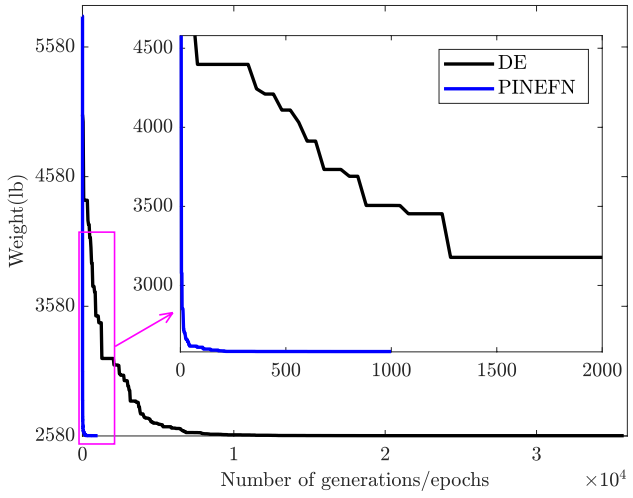

Fig. 8 The weight convergence histories of the 17-bar truss obtained using the PINEFN and DE

removed by our method, and only the neural network is used to perform the structural optimization task. Instead of solving the structural optimization, our approach minimizes the loss function to seek the optimal parameters of the network.

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \left(\mathcal{L}(\mathbf{X}, \boldsymbol{\theta}) \right). \tag{10}$$

In this work, Adam is a well-known gradient-based optimizer which is used to perform the training. Hence, the sensitivity analysis regarding computing the derivatives of the loss function with respect to each of all parameters $\nabla\mathcal{L}(\boldsymbol{\theta})$ must be determined. As mentioned above, this is easily and automatically calculated by utilizing BP which is a reverse mode of automatic differentiation and integrated into the common neural network with the Python library. For more details, interested readers are suggested to refer to Ref. [48]. As a result, the network parameters at epoch $(t+1)$ are adjusted as

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \frac{\mathbf{m}_{t+1}\sqrt{1-\beta_2^{(t+1)}}}{\left(1-\beta_1^{(t+1)}\right)\left(\sqrt{\mathbf{v}_{t+1}} + \varepsilon\sqrt{1-\beta_2^{(t+1)}}\right)}, \tag{11}$$

where $\mathbf{m}_{t+1}$ and $\mathbf{v}_{t+1}$ are obtained by

$$\begin{aligned} \mathbf{m}_{t+1} &= \beta_1 \mathbf{m}_t + \left(1-\beta_1\right).\nabla\mathcal{L}\left(\boldsymbol{\theta}_t\right), \\ \mathbf{v}_{t+1} &= \beta_2 \mathbf{v}_t + \left(1-\beta_2\right).\nabla\mathcal{L}\left(\boldsymbol{\theta}_t\right), \end{aligned} \tag{12}$$

in which $\boldsymbol{\theta}$ is the trainable parameter vector of the network that consists of weights and biases; $\mathbf{m}_{t+1}$ and $\mathbf{v}_{t+1}$ are the first and second raw moment vectors which are controlled by two exponential decay rates $\beta_1,\ \beta_2 \in [0,1)$; ε denotes a constant added to ensure numerical stability, and η is the learning rate. In this study, their default values as suggested by Kingma and Ba [47] were utilized to train the model. Interested readers can refer to [47] for more details. Finally, the above process which corresponded with steps 5-11 in

Table 8 Comparison of the obtained results for the 17-bar planar truss

A_i (in^2)	Lee [10]	Li [14]			Khot [2]	Adeli [12]	This study	
		PSO	PSOPC	HPSO			DE	PINEFN
A_1	15.821	15.766	15.981	15.896	15.93	16.029	15.920	15.906
A_2	0.108	2.263	0.1	0.103	0.1	0.107	0.100	0.113
A_3	11.996	13.854	12.142	12.092	12.07	12.183	12.075	12.068
A_4	0.1	0.106	0.1	0.1	0.1	0.11	0.100	0.100
A_5	8.15	11.356	8.098	8.063	8.067	8.417	8.070	8.074
A_6	5.507	3.915	5.566	5.591	5.562	5.715	5.554	5.557
A_7	11.829	8.071	11.732	11.915	11.933	11.331	11.936	11.917
A_8	0.1	0.1	0.1	0.1	0.1	0.105	0.100	0.100
A_9	7.934	5.85	7.982	7.965	7.945	7.301	7.940	7.949
A_{10}	0.1	2.294	0.113	0.1	0.1	0.115	0.100	0.103
A_{11}	4.093	6.313	4.074	4.076	4.055	4.046	4.055	4.061
A_{12}	0.1	3.375	0.132	0.1	0.1	0.101	0.100	0.100
A_{13}	5.66	5.434	5.667	5.67	5.657	5.611	5.668	5.665
A_{14}	4.061	3.918	3.991	3.998	4	4.046	3.988	4.005
A_{15}	5.656	3.534	5.555	5.548	5.558	5.152	5.561	5.550
A_{16}	0.1	2.314	0.101	0.103	0.1	0.107	0.101	0.104
A_{17}	5.582	3.542	5.555	5.537	5.579	5.286	5.583	5.579
W_{best} (lb)	2580.81	2724.37	2582.85	2581.94	2581.89	2594.42	2581.89	2581.97
CVE_{max}	0.04	None	None	None	None	1.69	None	None
Times (s)	–	–	–	–	–	–	268.96	6.73

Table 9 Error of the constraints for the 17-bar planar truss

Elements	Stress (ksi)			Dofs	Displacement (in)		
	FEA	PINEFN	Error (%)		FEA	PINEFN	Error (%)
1	25.0262	25.0270	0.0029	u_3	− 0.0834	− 0.0834	0.0024
2	24.1879	24.1961	0.0337	v_3	− 0.2447	− 0.2447	0.0214
3	− 25.0193	− 25.0187	0.0024	u_4	0.0834	0.0834	0.0029
4	− 1.6089	− 1.5939	0.9324	v_4	− 0.2500	− 0.2500	0.0009
5	24.9903	24.9884	0.0075	u_5	− 0.1668	− 0.1668	0.0016
6	24.9983	24.9995	0.0049	v_5	− 0.6669	− 0.6669	0.0015
7	− 25.0254	-25.0252	0.0008	u_6	0.1667	0.1667	0.0023
8	3.4540	3.4381	0.4608	v_6	− 0.6554	− 0.6555	0.0097
9	24.9810	24.9820	0.0041	u_7	− 0.2501	− 0.2501	0.0021
10	19.5849	19.6173	0.1657	v_7	− 1.2027	− 1.2030	0.0234
11	− 24.9757	− 24.9781	0.0096	u_8	0.2500	0.2500	0.0001
12	− 14.2641	− 14.1821	0.5746	v_8	− 1.2503	− 1.2503	0.0006
13	24.9641	24.9623	0.0070	u_9	− 0.3333	− 0.3333	0.0007
14	− 24.9688	− 24.9665	0.0092	v_9	− 2.0000	− 2.0000	0.0003
15	− 24.9889	− 24.9888	0.0001				
16	− 24.0933	− 24.0958	0.0105				
17	− 24.9873	− 24.9874	0.0004				
Max	25.0262	25.0270	0.9324	–	0.2500	0.2500	0.0234
Min	− 25.0254	− 25.0252	0.0001	–	− 2.0000	− 2.0000	0.0001

Algorithm 1 was repeated until the stop condition was satisfied. A pseudo-code of the training for the parameters tuning is summarized in Algorithm 1. And once the network is trained, the optimum weight is found at the minimum of the loss function corresponding to the optimal parameters without using any structural analyses.

Algorithm 1: Optimization of truss structures using PINEFN framework

Input:

- Structure: geometry, material properties, external forces, BCs
- Network: number of hidden layers, hidden neurons, activation function, Adam optimizer

Output: optimal parameters $\boldsymbol{\theta}$, optimum weight of truss structures

Collect the training data from the coordinates of truss members
Build a NN with uniformly distributed initial parameters $\boldsymbol{\theta}_0$ from [-1, 1]
Set the default values of Adam optimizer parameters [47]
while $\|\nabla \mathcal{L}(\boldsymbol{\theta})\| > 0.01$ or *Maxepoch* is not reached **do**
    Predict $\left(\hat{\mathbf{A}}(\mathbf{X}, \boldsymbol{\theta}), \hat{\mathbf{q}}(\mathbf{X}, \boldsymbol{\theta})\right)$ using the FF
    Compute the weight $W\left(\hat{\mathbf{A}}\right)$ of truss structure
    Evaluate the constraint functions $g_j\left(\hat{\mathbf{A}}, \hat{\mathbf{q}}\right)$ and residual $\mathbf{v}\left(\hat{\mathbf{A}}, \hat{\mathbf{q}}\right)$ by Eq. (9) based on the output values, physical laws, BCs, and external forces
    Loss function $\mathcal{L}(\boldsymbol{\theta}_t)$ is estimated by Eq. (8)
    $\frac{\partial \mathcal{L}}{\partial \theta_i}$ is calculated automatically by the BP algorithm
    Update parameters $\boldsymbol{\theta}_{t+1}$ of the network by Eq. (11)
    $t = t + 1$

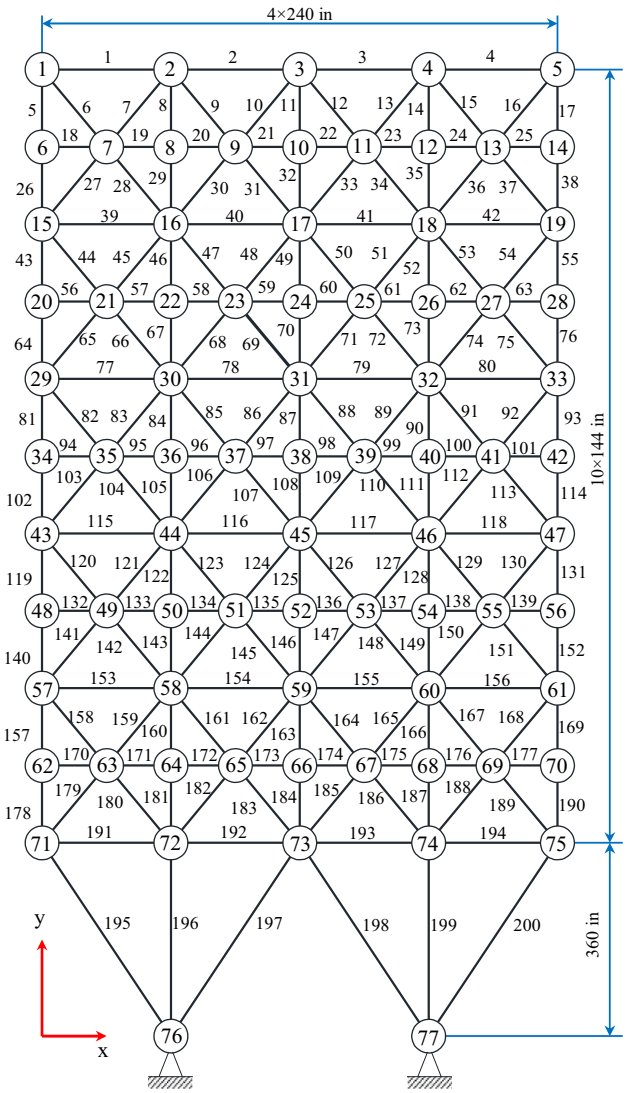

Fig. 9 A 200-bar planar truss structure

4 Numerical experiments

In the following, several numerical examples are investigated to demonstrate the efficiency of the presented approach for solving structural optimization problems. For this purpose, the obtained solutions will be compared with differential evolution algorithm (DE) and other methods achieved in the literature. In all examples, the LeakyReLU and Softmax are employed as the hidden and output activation functions, respectively. On the other hand, the other hyperparameters of the network for each problem which are found by Grid search and by trial-error methods are listed in Table 1. Note that the learning process is terminated when either the maximum number of epochs reaches or the norm of the gradient value is less than 0.01 [28, 37].

In addition, the parameters of the DE algorithm are selected as follows: population size 20, mutant factor $F = 0.8$, maximum number of evaluations 3000, crossover factor $Cr = 0.9$, and setting the value of convergence criteria to 10^{-6} [9, 23, 28]. Due to the DE algorithm's stochastic nature, the best solution to each problem is addressed by following 30 independent runs. All of the experiments were performed on a personal desktop computer using Python software.

4.1 Ten-bar truss

A ten-bar planar truss structure, as illustrated in Fig. 4, is examined as the first numerical example for design optimization. Cross-sectional areas of members are considered as design variables and their minimum values are specified as 0.1 in^2. The material density and Young's modulus are set as 0.1 lb/in^3 and 10^4 ksi, respectively. The displacement limits of ±2.0 in are applied to all free nodes. The allowable stress for all elements is restricted to ±25.0 ksi. In this example, two load cases are considered as following: (1) the first loading condition with $P_1 = 100$ kips and $P_2 = 0$; (2) the second loading condition with $P_1 = 150$ kips and $P_2 = 50$ kips. The architecture (4-30-30-2) and parameters of the network, as shown in Table 1, are employed to perform the training process.

A comparison of the optimal solution obtained by this study and previous works, including design variables, total weight of the structure (W), maximum constraint violation error (CVE), as well as computational time, is reported in Table 2 for the first loading case. Therein, CVE is expressed as follows

$$\mathrm{CVE}_j = \begin{cases} \text{none} & g_j \le 0, \\ \dfrac{\|g_j\|}{[\Delta]_j} 100\% & g_j > 0, \end{cases} \tag{13}$$

where CVE_j is the jth constraint violation error; g_j denotes the jth constraint value; $[\Delta]_j$ is the jth allowable displacement or stress.

Firstly, it is easily seen that the obtained result of the PINEFN (5061.06 lb) reveals a fairly good agreement with the DE (5060.86 lb), HPSO (5060.92 lb), and PSOPC (5061.00 lb). Note that even though the lightest design given by Lee (5057.88 lb) violated the design constraints (0.09%). Nevertheless, the PINEFN outperforms well-known existing algorithms in term of the solution quality as well as the computing times. More specifically, the proposed approach (5.02 s) only spends one-nineteenth of the computation costs of DE (96.52 s) to get the near-global optimal solution as possible. In addition, all constraints are satisfied without violation, and the weight error against the DE is very small with only 0.0039%. This can easily be explained by the fact that a sufficiently large number

Table 10 Design variables of the 200-bar planar truss

Design variables	Member group	Design variables	Member group
A_1	1, 2, 3, 4	A_{16}	82, 83, 85, 86, 88, 89, 91, 92, 103, 104, 106, 107, 109, 110, 112, 113
A_2	5, 8, 11, 14, 17	A_{17}	115, 116, 117, 118
A_3	19, 20, 21, 23, 24	A_{18}	119, 122, 125, 128, 131
A_4	18, 25, 56, 63, 94,101, 132, 139, 170, 177	A_{19}	133, 134, 135, 136, 137, 138
A_5	26, 29, 32, 35, 38	A_{20}	140, 143, 146, 149, 152
A_6	6, 7, 9, 10, 12, 13, 15,16, 27, 28, 30, 31, 33,34, 36, 37	A_{21}	120, 121, 123, 124, 126, 127,129, 130, 141, 142, 144, 145, 147, 148, 150, 151
A_7	39, 40, 41, 42	A_{22}	153, 154, 155, 156
A_8	43, 46, 49, 52, 55	A_{23}	157, 160, 163, 166, 169
A_9	57, 58, 59, 60, 61, 62	A_{24}	171, 172, 173, 174, 175, 176
A_{10}	64, 67, 70, 73, 76	A_{25}	178, 181, 184, 187, 190
A_{11}	44, 45, 47, 48, 50, 51,53, 54, 65, 66, 68, 69,71, 72, 74, 75	A_{26}	158, 159, 161, 162, 164, 165, 167, 168, 179, 180, 182, 183, 185, 186, 188, 189
A_{12}	77, 78, 79, 80	A_{27}	191, 192, 193, 194
A_{13}	81, 84, 87, 90, 93	A_{28}	195, 197, 198, 200
A_{14}	95, 96, 97, 98, 99, 100	A_{29}	196, 199
A_{15}	102, 105, 108, 111, 114		

of independent runs are demanded for the metaheuristic algorithms due to their stochastic nature [4, 7, 10, 14]. Meanwhile, our training model relied on the gradient descent algorithm, so it only requires one run time and saves a massive amount of the computational cost. And no structural analysis is of special importance here. In addition to evaluate the effect of the PINEFN, the constraint values gained through this work are compared with the exact values attained by FEA using the obtained optimal solution reported in Table 3. Obviously, none of the stress and displacement constraints are violated. Furthermore, it can be observed that the structural responses acquired by PINEFN are very close to FEA with the error less than 0.04%. Besides, the weight convergence histories of two algorithms are illustrated in Fig. 5. Clearly, our procedure converges very fast at the beginning, tends to a fairly stable performance at the epoch of 500 approximately, and then arrives at the near-optimal solution only through 1000 epochs. More importantly, it works completely without using FEA. By contrast, the DE algorithm converges very slowly and demands a large number of FEA evaluations (18,041) to reach the optimum weight.

According to evaluate the impact of various hidden activation functions and optimizers on the performance of the network, a survey is carried out to determine the most suitable combination based on the network architecture (4-30-30-2). Herein, the mean square error (MSE) of the optimum weight between the PINEFN and DE is used as a standard measurement tool. And the obtained results are summarized in Table 4. Evidently, Adam and LeakyReLU are the best optimizer and activation functions, respectively. Therein, their combination yields the lowest MSE (0.0039%), so it is chosen for the training phase. At the same time, the grid search procedure is applied to select the number of hidden neurons and layers. MSEs of the optimum weight are shown in Table 5 for each case. From the data in this table, it can be seen that increasing the number of hidden neurons and layers cannot always improve the accuracy of the model. In this example, the network architecture with two hidden layers and thirteen neurons is the most suitable with the smallest MSE (0.0039%).

For case 2, the optimal results obtained by PINEFN, and previous works are illustrated in Tables 6, 7 and Fig. 6. Similarly, in case 1, the optimum weight found by the PINEFN (4677.25 lb) is close to Rizzi [4](4676.92 lb), DE (4676.93 lb), and smaller than the other studies (Schmit [7]: 4691.84 lb; PSO [14]: 4679.47 lb; PSOPC [14]: 4677.70 lb; and HPSO [14]: 4677.29 lb). Again, our framework takes only 5.83 s, while DE requires 104.29 s to get as accurate solutions as possible. And clearly, it saves more than 17 times the computational cost. As can be seen from the data of Table 7, it is obvious that all constraints are not violated and the largest error value (0.7538%) is less than 1%. Again, PINEFN can achieve comparable accuracy compared to traditional methods. In addition, PINEFN converges very quickly to the optimal solution, while the DE is still a long way from the desired value.

Table 11 Optimization results obtained for the 200-bar planar truss

A_i (in^2)	Lee [12]	Kaveh [20]	Lamberti [49]	Degertekin [11]	Degertekin [22]	Pierezan [19]	Mai [28]		PINEFN
							DE	DUL	
A_1	0.1253	0.1033	0.1468	0.150	0.146	0.1390	0.1206	0.1183	0.1032
A_2	1.0157	0.9184	0.9400	0.946	0.941	0.9355	0.9345	0.9907	0.9470
A_3	0.1069	0.1202	0.1000	0.101	0.100	0.1000	0.1168	0.1142	0.1158
A_4	0.1096	0.1009	0.1000	0.100	0.101	0.1000	0.1000	0.2323	0.1111
A_5	1.9369	1.8664	1.9400	1.945	1.941	1.9355	1.9292	1.9580	1.9501
A_6	0.2686	0.2826	0.2962	0.296	0.296	0.2909	0.2870	0.2906	0.2977
A_7	0.1042	0.1000	0.1000	0.100	0.100	0.1000	0.1102	0.1624	0.1229
A_8	2.9731	2.9683	3.1042	3.161	3.121	3.0816	3.0780	3.1520	3.1354
A_9	0.1309	0.1000	0.1000	0.102	0.100	0.1000	0.2074	0.1381	0.1036
A_{10}	4.1831	3.9456	4.1042	4.199	4.173	4.0816	4.0783	4.1834	4.1312
A_{11}	0.3967	0.3742	0.4034	0.401	0.401	0.3967	0.4329	0.3844	0.4214
A_{12}	0.4416	0.4501	0.1912	0.181	0.181	0.2959	0.1546	0.2115	0.1057
A_{13}	5.1873	4.9603	5.4284	5.431	5.423	5.3854	5.3500	5.4466	5.4590
A_{14}	0.1912	1.0738	0.1000	0.100	0.100	0.1000	0.1027	0.1273	0.1056
A_{15}	6.2410	5.9785	6.4284	6.428	6.422	6.3853	6.3502	6.4490	6.4572
A_{16}	0.6994	0.7863	0.5734	0.571	0.571	0.6332	0.5636	0.5317	0.5505
A_{17}	0.1158	0.7374	0.1327	0.156	0.156	0.1842	0.5160	0.2150	0.1232
A_{18}	7.7643	7.3809	7.9717	7.961	7.958	8.0396	7.9508	8.0113	7.9973
A_{19}	0.1000	0.6674	0.1000	0.100	0.100	0.1000	0.1017	0.1434	0.1391
A_{20}	8.8279	8.3000	8.9717	8.959	8.958	9.0395	8.9503	8.9967	8.9847
A_{21}	0.6986	1.1967	0.7049	0.722	0.720	0.7460	0.8932	0.7054	0.7276
A_{22}	1.5563	1.0000	0.4196	0.491	0.478	0.1306	0.1525	0.2450	0.2385
A_{23}	10.9806	10.8262	10.8636	10.909	10.897	10.9114	11.0423	10.8275	10.8581
A_{24}	0.1317	0.1000	0.1000	0.101	0.100	0.1000	0.1000	0.1266	0.1680
A_{25}	12.1492	11.6976	11.8606	11.985	11.897	11.9114	12.0423	11.8557	11.8564
A_{26}	1.6373	1.3880	1.0339	1.084	1.080	0.8627	0.9196	0.8580	0.9799
A_{27}	5.0032	4.9523	6.6818	6.464	6.462	6.9169	6.7136	6.8944	6.6773
A_{28}	9.3545	8.8000	10.8113	10.802	10.799	10.9674	10.7305	11.1690	10.9118
A_{29}	15.0919	14.6645	13.8404	13.936	13.922	13.6742	13.8833	13.6032	13.7673
W_{best} (lb)	25447.10	25156.5	25445.63	25542.5	25488.15	25450.18	25564.99	25547.90	25436.49
CVE_{max}	3.69	9.97	0.071	None	None	None	None	None	None
Times (s)	–	–	–	–	–	–	3553.64	1305.27	100.91

4.2 Seventeen-bar truss

A seventeen-bar planar truss structure is investigated as the second example for size optimization. The geometry, loading, element representation, and boundary condition are schematized in Fig. 7. The linear elastic modulus and material density are 30,000 ksi and 0.268 lb/in^3, respectively. Cross-sectional areas of members are considered as design variables. All displacement of nodes are restricted to ±2.0 in, and stress limitations of members ±50 ksi were imposed on all members. The minimum cross-sectional areas are set at 0.1 in^2. The network architecture (4-40-40-2) is used to train this problem with 1,000 epochs.

As the first presented example, the optimal solutions achieved by this study in comparison with other works are tabulated in Table 8. It is obvious that the optimum weight found by the PINEFN (2581.97 lb) agrees well with Khot [2] (2581.89 lb), DE [2] (2581.89 lb), and HPSO [14] (2581.94 lb) without violating constraints. However, it is clear that our model outperforms the metaheuristic algorithms in terms of computation times. In particular, its computation cost (6.73 s) reduces more than 39 times compared with the DE (268.96 s). All constraints and their error are summarized in Table 9 which shows a comparison of PINEFN and FEA using the obtained optimal cross-sectional areas. And it is easily seen that all the optimized displacements and stresses are free from any violations of constraints. In addition, the overall structural responses found by the PINEFN are similar to that of the FEA with the error less than 1%. A comparison of the convergence history between the proposed

Table 12 Error of the constraints for the 200-bar planar truss

Elements	Stress (ksi)		
	FEA	PINEFN	Error (%)
17	0.0000	0.0000	1.2367
19	0.0000	0.0000	0.9818
41	− 0.3415	− 0.3418	0.0774
64	− 9.6038	− 9.6038	0.0000
86	9.4741	9.4741	0.0000
101	− 9.9995	− 9.9994	0.0009
121	9.9782	9.9783	0.0017
122	9.9070	9.9067	0.0028
157	9.9770	9.9768	0.0017
173	− 10.0000	− 10.0000	0.0002
Max	9.9782	9.9783	1.2367
Min	− 10.0000	− 10.0000	0.0000

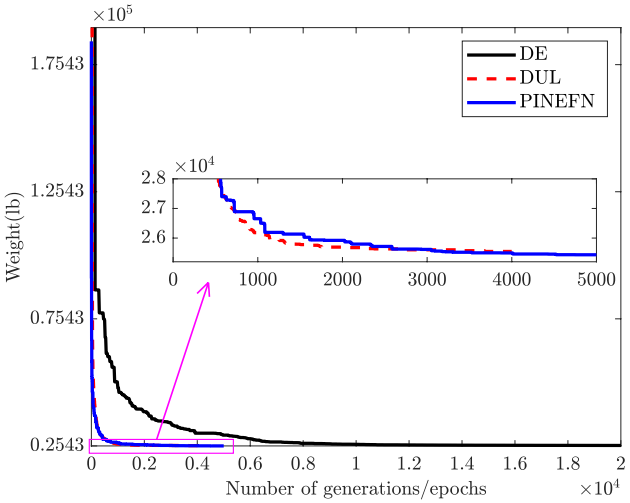

Fig. 10 The weight convergence histories of the 200-bar truss obtained using the PINEFN and other algorithms

method and DE is illustrated in Fig. 8. Clearly, the PINEFN converges more rapidly than the DE. It only requires 1,000 epochs without FEA, while the DE uses 35,761 FEA simulations to gain the near-global optimal weight.

4.3 200-bar planar truss

The next optimization problem deals with the 200-bar planar truss shown in Fig. 9. All members are divided into 29 groups corresponding to the design variables which are indicated in Table 10. The material density and linear elastic modulus are 0.283 lb/in^3 and 30,000 ksi for all elements. This structure is subjected to stress limitations of ±10 ksi. The minimum design variable is 0.1 in^2. It is designed for three loading cases: 1) 1 kip acting in the positive direction of the x-axis at nodes 1, 6, 15, 20, 29, 34, 43, 48,57, 62 and 71; (2) 10 kip applying in the negative direction of the y-axis at nodes 1, 2, 3, 4, 5, 6, 8, 10, 12, 14, 15, 16, 17, 18, 19, 20,22, 24, 26, 28, 29, 30, 31, 32, 33, 34, 36, 38, 40, 42, 43, 44, 45, 46, 47, 48, 50, 52, 54, 56, 57, 58, 59, 60, 61, 62, 64, 66, 68, 70, 71, 72, 73, 74 and 75; (3) including conditions (1) and (2) acting together. In this example, the network configuration with 3 hidden layers, 40 neurons per hidden layer, and 5,000 epochs are carried out to perform the training process.

The optimum weight and design variables are reported in Table 11. It is worth mentioning that the PINEFN attains the lightest design overall because Kaveh's [20] lighter design violates the design constraints. The results shown that our model (25436.49 lb) can decrease the structural weight by about 13 lb compared with the second best weight obtained by Pierezan [19] (25450.18 lb). It is interesting here that PINEFN outperforms the state-of-the-art unsupervised approach DUL (25547.90 lb) by Mai [28] in terms of both the quality of solution and computation cost. It is worth noting that, for the DUL, the neural network as the optimal tool directly participated in the optimization process, but FEA is still required to determine the structural responses. Meanwhile, this numerical simulation was completely removed by PINEFN. Clearly, the computation time of PINEFN was reduced by more than 13 and 35 times compared with the DUL and DE, respectively. Furthermore, this leads to a significant reduction in computational effort with the increasing structural complexity. Several maximum and minimum values of constraints as well as their error are illustrated in Table 12. It is easily seen that the stresses found by the PINEFN agree well with the FEA and satisfy allowable stress constraint. Note that although the error of the 17th member (1.2367%) is a little bigger than 1%, its stress value (0.0000 ksi) is very small compared to the the stress

Table 13 Loading conditions for the 25-bar space truss (kips)

Node	Case 1			Case 2		
	F_x	F_y	F_z	F_x	F_y	F_z
1	0.0	20.0	-5.0	1.0	10.0	− 0.5
2	0.0	− 20.0	− 5.0	0.0	10.0	− 0.5
3	0.0	0.0	0.0	0.5	0.0	0.0
6	0.0	0.0	0.0	0.5	0.0	0.0

Table 14 Stress limitation for the 25-bar space truss

A_i (in^2)	Allowable compressive stress (ksi)	Allowable tension stress (ksi)
A_1	35.092	40.0
A_2-A_5	11.590	40.0
A_6-A_9	17.305	40.0
A_{10}-A_{11}	35.092	40.0
A_{12}-A_{13}	35.092	40.0
A_{14}-A_{17}	6.7590	40.0
A_{18}-A_{21}	6.9590	40.0
A_{22}-A_{25}	11.082	40.0

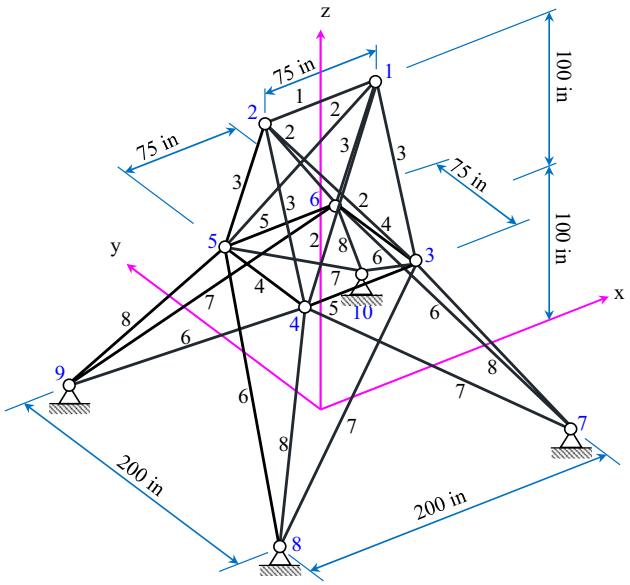

Fig. 11 A 25-bar space truss structure

limitation (10 ksi). The convergence histories in Fig. 10 provide a more detailed view of three different methods. As can be seen on the plot, the convergence rates of PINEFN and DUL are similar in shape and rapidly decrease at the beginning. It can be easily explained by the fact that both approaches utilized the neural network based on the gradient descent algorithm. However, the proposed method converged better than the DUL at the training end. And this difference comes from the following reasons: (i) The sensitivity of the DUL is calculated by the BP and automatic differentiation tool JAX, while the PINEFN only uses the BP; (ii) the structural responses obtained by the approximate solution of the FEA for the DUL whereas they are directly determined from the output values of the network for the present method. And this example is no exception to the DE algorithm which is the slowest convergence rate.

4.4 25-bar space truss

A twenty five-bar spatial truss, as shown in Fig. 11, is examined as the next design optimization with stress and displacement constraints. The modulus of elasticity is 10,000 ksi and the density is 0.1 lb/in^3 for all members. Two loading cases given in Table 13 are considered for this structure. All members are categorized into 8 groups corresponding to the design variables as listed in Table 14. All displacements of nodes are restricted in the interval [- 0.35, 0.35] in. In addition, the allowable stresses of members are listed in Table 14. To find the optimum weight, the network configuration shown in Table 1 is used to train the model.

As previously investigated examples, the results obtained by the presented work and the other algorithms recently published in the literature are summarized in Table 15. It is obvious from the data in this table that the optimum design obtained by PINEFN (545.22 lb) shows a strong agreement

Table 15 Optimization results obtained for the 25-bar space truss

A_i (in^2)	Lee [10]	Li [14]	Kaveh [20]	Degertekin [22]	Mai [28]		Camp [21]	PINEFN
					DE	DUL		
A_1	0.047	0.010	2.662	0.0100	0.0100	0.0131	0.010	0.0127
A_2-A_5	2.022	1.970	1.993	2.0712	1.9834	1.9515	2.092	1.9809
A_6-A_9	2.950	3.016	3.056	2.9570	2.9984	2.9662	2.964	3.0039
A_{10}-A_{11}	0.010	0.010	0.010	0.0100	0.0100	0.0125	0.010	0.0112
A_{12}-A_{13}	0.014	0.010	0.010	0.0100	0.0100	0.0128	0.010	0.0114
A_{14}-A_{17}	0.688	0.694	0.665	0.6891	0.6864	0.6968	0.689	0.6865
A_{18}-A_{21}	1.657	1.681	1.642	1.6209	1.6776	1.7260	1.601	1.6771
A_{22}-A_{25}	2.663	2.643	2.679	2.6768	2.6576	2.6430	2.686	2.6560
W_{best} (lb)	544.38	545.19	545.16	545.09	545.16	545.71	545.38	545.22
CVE_{max}	0.206	None	2.06	None	None	None	None	None
Times (s)	–	–	–	–	104.07	49.47	–	6.68

Table 16 Error of the constraints for the 25-bar space truss

Elements	Stress (ksi)			Dofs	Displacement (in)		
	FEA	PINEFN	Error (%)		FEA	PINEFN	Error (%)
1	5.2070	5.2056	0.0263	u_1	− 0.0195	− 0.0195	0.0263
2	− 7.0015	− 7.0016	0.0026	v_1	0.3500	0.3500	0.0004
3	6.9434	6.9434	0.0000	w_1	− 0.0289	− 0.0289	0.0045
6	− 6.5914	− 6.5913	0.0009	u_2	0.0195	0.0195	0.0263
7	4.8451	4.8451	0.0010	v_2	− 0.3500	− 0.3500	0.0004
10	− 1.7707	− 1.7702	0.0273	w_2	− 0.0289	− 0.0289	0.0045
12	− 1.8416	− 1.8410	0.0323	u_3	0.1116	0.1116	0.0025
14	− 2.6125	− 2.6124	0.0018	v_3	− 0.0403	− 0.0403	0.0046
15	1.0157	1.0158	0.0100	w_3	− 0.1000	− 0.1000	0.0008
18	5.3284	5.3285	0.0018	u_4	0.0978	0.0978	0.0017
19	− 6.9577	− 6.9575	0.0015	v_4	0.0270	0.0270	0.0065
22	− 1.1649	− 1.1649	0.0009	w_4	0.0573	0.0573	0.0034
Max	6.9434	6.9434	0.0323	–	0.3500	0.3500	0.0263
Min	− 7.0015	− 7.0016	0.0000	–	− 0.3500	− 0.3500	0.0004

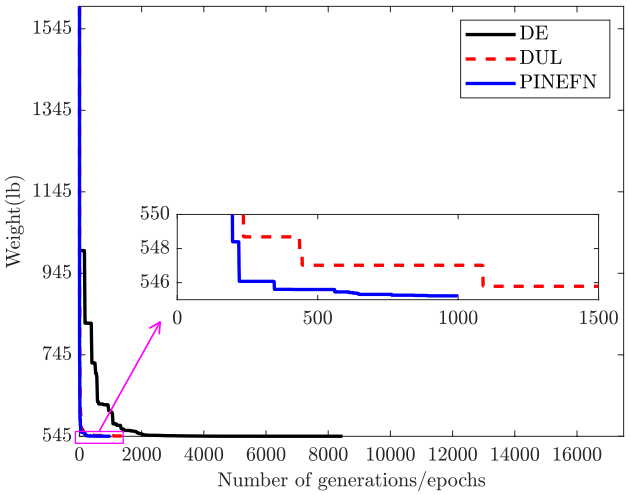

Fig. 12 The weight convergence histories of the 25-bar truss obtained using the PINEFN and other algorithms

with Degertekin [22] (545.09 lb), DE [28] (545.16 lb), Li [14] (545.19 lb), and smaller than the other studies (DUL [28]: 545.71 lb; Camp [21]: 545.38 lb). Although the obtained results by Lee [10] (544.38 lb) and Kaveh [20] (545.16 lb) are relatively lighter than our method, the design constraints are violated. Furthermore, our model saved more than 15 and 7 times the computation cost compared to DE and DUL, respectively. Table 16 shows several constraints and their error with the FEA. Clearly, no any violations of constraints are observed here. In addition, the constraint error against to the FEA is less than 0.4%. And this once again proves the efficiency of the proposed work. A comparison of the convergence curve between the PINEFN, DUL, and DE is illustrated in Fig. 12. It is easily seen that the proposed framework always converges much more rapidly than the DE and DUL. It only requires 1000 epochs, while the DUL and DE require 1500 and 8480 times of linear analysis.

4.5 72-bar space truss

The next example deals with the design of 72-bar four-level skeletal tower, and the cross-sectional areas are categorized into 16 groups as illustrated in Fig. 13. All members are made of material having density of 0.1 lb/in^3, elasticity modulus of 10,000 ksi, and allowable stress of ±25 ksi. In addition, all displacements of nodes are restricted to ±0.25 in. It is subjected to two loading conditions, as listed in Table 17. Therein, the minimum design variables are defined as 0.1 in^2 and 0.01 in^2 for cases 1 and 2, respectively. A network architecture with three hidden layers, 60 neurons per hidden layer, and 1,000 epochs is the finest performance for this application.

Tables 18 and 20 provide a comparison between the obtained result of the PINEFN and other available algorithms in the literature for the optimal solution. In both cases, it is easily seen that the PINEFN found the lightest design with the least time and effort overall without violating constraints. More specifically, for the first case, it only takes 8.93 s to gain the optimal solution, while DUL and DE need 63.71 s and 488.14 s. Whereas its computational overhead (18.46 s) is 129 times less than that of the DE (2410.19 s) for the second case. Therefore, the present approach has proven again to be the most efficient for improving accuracy and reducing the computational effort. Besides, Tables 19 and 21 provide comparisons of constraints found by the PINEFN and FEA using the obtained optimal solution. As the previously presented examples, the constraint results agree well with the FEA with the error less than 0.4%. Finally, Figs. 14 and 15 depict the weight convergence histories of the present

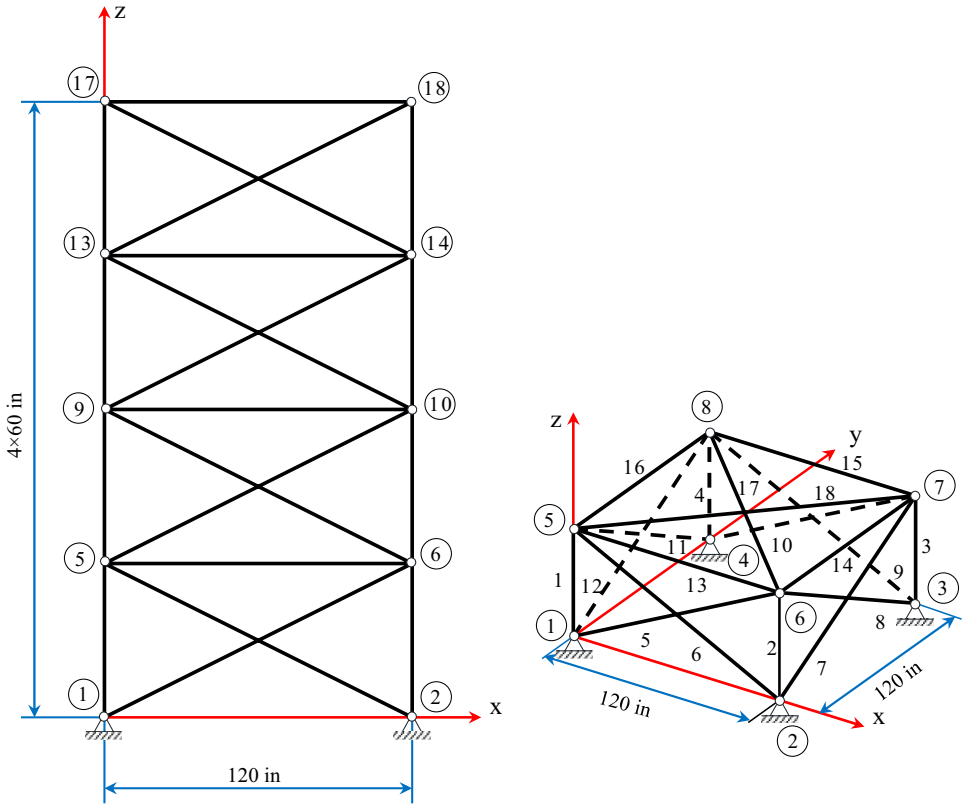

Fig. 13 A 72-bar space truss structure

Table 17 Loading conditions for the 72-bar space truss (kips)

Node	Case 1			Case 2		
	F_x	F_y	F_z	F_x	F_y	F_z
17	5.0	5.0	− 5.0	0.0	0.0	− 5.0
18	0.0	0.0	0.0	0.0	0.0	− 5.0
19	0.0	0.0	0.0	0.0	0.0	− 5.0
20	0.0	0.0	0.0	0.0	0.0	− 5.0

method, DUL, and DE. As the previously discussed examples, the convergence speed of the learning process is faster than the other studies, and the optimal design is found only after 1,000 epochs.

4.6 120-bar dome truss

To demonstrate the practical capability of the PINEFN, a 120-bar dome truss structure was considered the last optimal design problem. Its configuration and dimensions can be found in Fig. 16. As shown in this plot, the cross-sectional areas of members, which are design variables, are classified into seven groups. The density, elasticity modulus (E), and yield stress (σ_y) of the steel material for all members are taken as 0.288 lb/in^3, 30,450 ksi, and 58.0 ksi, respectively. According to the AISC ASD (1989) [53], the allowable tensile and compressive stresses are estimated as follows:

$$\begin{cases} \sigma_i^u = 0.6\sigma_y & \text{for } \sigma_i \geq 0 \\ \sigma_i^l & \text{for } \sigma_i < 0 \end{cases} \tag{14}$$

$$\sigma_i^l = \begin{cases} \left[\left(1 - \frac{\lambda_i^2}{2C_C^2}\right)\sigma_y\right] \Big/ \left(\frac{5}{3} + \frac{3\lambda_i}{8C_C} - \frac{\lambda_i^3}{8C_C^3}\right) & \text{for } \lambda_i < C_C \\ \frac{12\pi^2 E}{23\lambda_i^2} & \text{for } \lambda_i \geq C_C \end{cases} \tag{15}$$

where λ_i denotes the slenderness ratio ($\lambda_i = kL_i/r_i$); L_i is the member length; k is the effective length factor; r_i is the radius of gyration ($r_i = aA_i^b$); a and b are the constants that depend on the types of section of the members, and pipe section, with values 0.4993 and 0.6777, are used for bars, respectively; C_C expresses the slenderness factor splitting the elastic and inelastic buckling regions ($C_C = \sqrt{2\pi^2 E/\sigma_y}$). The system is subjected to vertical loads in the negative

Table 18 Optimization results obtained for the 72-bar space truss (Case 01)

A_i (in^2)	Lee [10]	Camp [21]	Li [14]	Kaveh [20]	Degertekin [22]	Bekdaş [50]	Ehsan [51]	Mai [28]		PINEFN
								DE	DUL	
A_1-A_4	1.790	1.8577	1.857	1.9042	1.9064	1.8758	1.9298	1.8356	1.8606	1.8570
A_5-A_{12}	0.521	0.5059	0.505	0.5162	0.5061	0.5160	0.5090	0.5359	0.5008	0.5047
A_{13}-A_{16}	0.100	0.1000	0.100	0.1000	0.1000	0.1000	0.1000	0.1002	0.1015	0.1000
A_{17}-A_{18}	0.100	0.1000	0.100	0.1000	0.1000	0.1000	0.1000	0.1001	0.1012	0.1000
A_{19}-A_{22}	1.229	1.2476	1.255	1.2582	1.2617	1.2993	1.2467	1.2991	1.2635	1.2542
A_{23}-A_{30}	0.522	0.5269	0.503	0.5035	0.5111	0.5246	0.5128	0.4959	0.5061	0.5037
A_{31}-A_{34}	0.100	0.1000	0.100	0.1000	0.1000	0.1001	0.1000	0.1001	0.1014	0.1000
A_{35}-A_{36}	0.100	0.1012	0.100	0.1000	0.1000	0.1000	0.1000	0.1007	0.1010	0.1000
A_{37}-A_{40}	0.517	0.5209	0.496	0.5178	0.5317	0.4971	0.5298	0.4759	0.4971	0.4954
A_{41}-A_{48}	0.504	0.5172	0.506	0.5214	0.5159	0.5089	0.5172	0.5140	0.5078	0.5078
A_{49}-A_{52}	0.100	0.1004	0.100	0.1000	0.1000	0.1000	0.1000	0.1000	0.1018	0.1002
A_{53}-A_{54}	0.101	0.1005	0.100	0.1007	0.1000	0.1000	0.1000	0.1043	0.1032	0.1004
A_{55}-A_{58}	0.156	0.1565	0.100	0.1566	0.1562	0.1575	0.1564	0.1002	0.1003	0.1000
A_{59}-A_{66}	0.547	0.5507	0.524	0.5421	0.5493	0.5329	0.5440	0.4932	0.5186	0.5223
A_{67}-A_{70}	0.442	0.3922	0.400	0.4132	0.4097	0.4089	0.4106	0.3840	0.4013	0.3984
A_{71}-A_{72}	0.590	0.5922	0.534	0.5756	0.5698	0.5731	0.5624	0.5658	0.5375	0.5356
W_{best} (lb)	379.27	379.85	369.65	379.66	379.63	379.10	379.65	370.30	370.04	369.66
CVE_{max}	0.218	None	39.075	None	None	None	None	None	None	None
Times (s)	–	–	–	–	–	–	–	488.14	63.72	8.93

Table 19 Error of the constraints for the 72-bar planar truss (Case 01)

Elements	Stress (ksi)			Dofs	Displacement (in)		
	FEA	PINEFN	Error (%)		FEA	PINEFN	Error (%)
16	0.2043	0.2041	0.0937	u_5	0.0173	0.0173	0.0011
21	− 3.8084	− 3.8084	0.0001	u_6	0.0400	0.0400	0.0131
43	− 2.4993	− 2.4993	0.0002	v_6	0.0400	0.0400	0.0131
45	− 2.4993	− 2.4993	0.0002	w_8	0.1209	0.1209	0.0012
49	4.1340	4.1327	0.0317	v_9	0.1209	0.1209	0.0012
52	4.1340	4.1327	0.0317	w_9	− 0.0751	− 0.0751	0.0055
54	5.4555	5.4538	0.0320	u_{15}	− 0.1168	− 0.1167	0.0399
55	− 23.8231	− 23.8151	0.0336	u_{17}	− 0.1168	− 0.1167	0.0399
56	0.2181	0.2176	0.2371	v_{17}	0.2500	0.2500	0.0018
57	− 7.1948	− 7.1925	0.0317	w_{17}	0.2500	0.2500	0.0018
58	0.2181	0.2176	0.2371	u_{19}	− 0.1183	− 0.1183	0.0081
60	− 5.6034	− 5.6029	0.0095	v_{19}	0.2133	0.2133	0.0031
Max	5.4555	5.4538	0.2371	–	0.2500	0.2500	0.0399
Min	− 23.8231	− 23.8151	0.0001	–	− 0.1183	− 0.1183	0.0011

direction of the z-axis which are 13.49 kips at node 1, 6.744 kips at nodes 2-13, and 2.248 kips at nodes 14-37. In addition to the stress constraints, the vertical displacement of free nodes under this loading is limited to 0.1969 in. The minimum values of the design variables are 0.775 in^2. To reach the goal, a network (4-60-60-60-2) with 1,000 epochs is chosen to perform the training.

A comparison of the results obtained by the present approach and other methods is reported in Tables 22 and 23. As expected, our obtained result (32,505.93 lb) is much

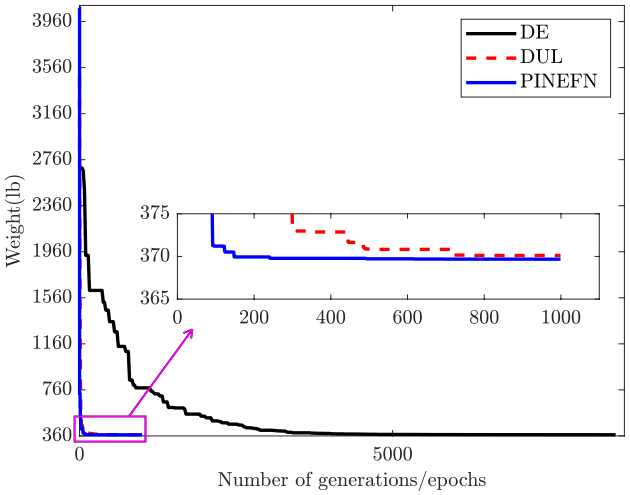

Fig. 14 The weight convergence histories of the 72-bar truss obtained using the PINEFN and other algorithms for the first load case

better than other studies (Kaveh [15]: 33,250.05 lb; Talatahari [16]: 33,251.22 lb; Adil [54]: 33,249.75 lb) without violating constraints. Furthermore, the PINEFN obtained the minimum weight only after 20.89 s, while the DE algorithm gained the same weight (32,499.65 lb) with 2,311.08 s. Once again, our model saves a massive amount of computational effort, but yet still delivering high-quality optimal solutions, and the error constraints of PINEFN are less than 0.01%. In addition, a comparison of the convergence rates is shown in Fig. 17. The mass of structures rapidly decreases in the first 100 epochs and finds the solution only through 1,000 epochs. Whereas the DE algorithm is still a long way from the target value. The above results have proven the efficiency and robustness of the proposed approach for the structural optimization without any numerical simulations.

5 Conclusions

In this article, a physics-informed neural energy-force network framework has been successfully developed for solving the design optimization of truss structures. Its outstanding characteristic is that the structural analysis is purely removed during the optimization process, and only the neural network is built based on the physics laws to find the optimal structure. Herein, the self-adaptive penalty function, known as the loss function, is derived based

Table 20 Optimization results obtained for the 72-bar space truss (Case 02)

A_i (in^2)	Adeli [52]	Adeli [31]	Sarma [13]	Lee [10]	Li [14]	This study	
						DE	PINEFN
A_1-A_4	2.0259	2.755	1.732	1.963	1.907	1.8909	1.8345
A_5-A_{12}	0.5332	0.51	0.522	0.481	0.524	0.5195	0.5203
A_{13}-A_{16}	0.0100	0.01	0.01	0.01	0.01	0.0100	0.0100
A_{17}-A_{18}	0.01	0.01	0.013	0.011	0.01	0.0100	0.0101
A_{19}-A_{22}	1.157	1.37	1.345	1.233	1.288	1.2914	1.3136
A_{23}-A_{30}	0.569	0.507	0.551	0.506	0.523	0.5199	0.5176
A_{31}-A_{34}	0.01	0.01	0.01	0.011	0.01	0.0111	0.0100
A_{35}-A_{36}	0.01	0.01	0.013	0.012	0.01	0.0185	0.0109
A_{37}-A_{40}	0.514	0.481	0.492	0.538	0.544	0.5225	0.5214
A_{41}-A_{48}	0.479	0.508	0.545	0.533	0.528	0.5187	0.5188
A_{49}-A_{52}	0.01	0.01	0.066	0.01	0.019	0.0100	0.0101
A_{53}-A_{54}	0.01	0.064	0.013	0.167	0.02	0.1092	0.1014
A_{55}-A_{58}	0.158	0.215	0.178	0.161	0.176	0.1769	0.1681
A_{59}-A_{66}	0.55	0.518	0.524	0.542	0.535	0.5361	0.5377
A_{67}-A_{70}	0.345	0.419	0.396	0.478	0.426	0.4491	0.4527
A_{71}-A_{72}	0.498	0.504	0.595	0.551	0.612	0.5788	0.5838
W_{best} (lb)	379.3	376.5	364.4	364.33	364.86	365.30	364.05
CVE_{max}	–	–	–	–	–	None	None
Times (s)	–	–	–	–	–	2410.19	18.46

Table 21 Error of the constraints for the 72-bar space truss (Case 02)

Elements	Stress (ksi)			Dofs	Displacement (in)		
	FEA	PINEFN	Error (%)		FEA	PINEFN	Error (%)
2	− 0.9253	− 0.9253	0.0001	u_8	− 0.0056	− 0.0056	0.0001
4	− 0.9253	− 0.9253	0.0001	u_{14}	− 0.0232	− 0.0232	0.0066
34	− 2.5996	− 2.5996	0.0001	u_{15}	− 0.0771	− 0.0771	0.0023
39	− 5.1629	− 5.1630	0.0019	v_{15}	0.1440	0.1440	0.0001
49	4.9770	4.9775	0.0091	w_{15}	0.1440	0.1440	0.0001
50	− 0.0569	− 0.0571	0.3693	u_{17}	− 0.0783	− 0.0783	0.0017
51	− 0.0569	− 0.0571	0.3693	v_{17}	0.2500	0.2500	0.0011
52	4.9770	4.9775	0.0091	w_{17}	0.2500	0.2500	0.0011
53	− 0.5119	− 0.5115	0.0718	u_{18}	− 0.0267	− 0.0267	0.0091
54	5.4321	5.4320	0.0023	u_{19}	-0.1059	− 0.1059	0.0007
55	− 16.8499	− 16.8501	0.0015	v_{19}	0.2115	0.2115	0.0004
57	− 4.7969	− 4.7967	0.0037	u_{20}	− 0.0267	− 0.0267	0.0091
Max	5.4321	5.4320	0.3693	–	0.2500	0.2500	0.0091
Min	− 16.8499	− 16.8501	0.0001	–	− 0.1059	− 0.1059	0.0001

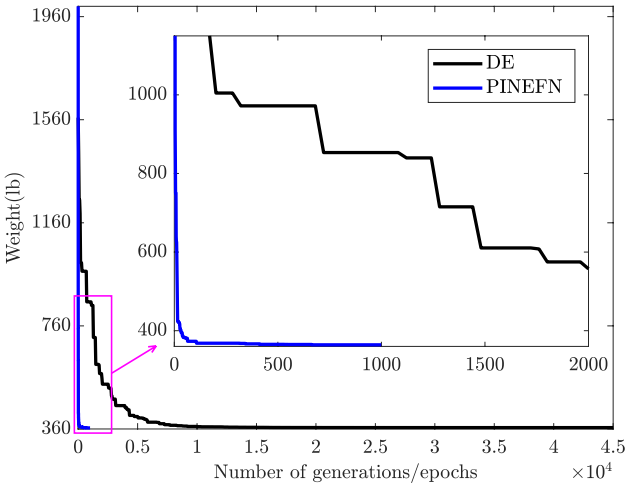

Fig. 15 The weight convergence histories of the 72-bar truss obtained using the PINEFN and DE for the second load case

on the weight, complementary energy, and constitutive equations. Accordingly, the optimum weight of the structure is found at the end of the training when achieving the minimum loss. The robustness, efficiency, and reliability of the proposed framework are demonstrated through several benchmarks for size optimization of truss structures. Numerical results have indicated that the optimum weight obtained by this study outperforms previously released works in terms of the quality solution, convergence speed, and computational cost. Furthermore, the PINEFN is also significant because we can easily perform the structural optimization without using any structural analyses. In addition, one of the interesting things about this paradigm is that its learning possibility only relies upon the set of nodal coordinates which are known as the input data.

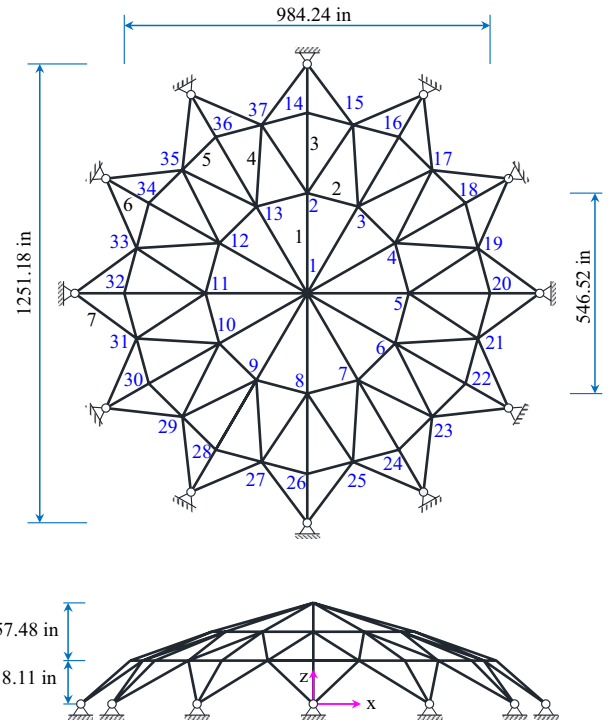

Fig. 16 A 120-bar dome truss structure

Hence, the obtained results as well as the whole training data independent of the sampling techniques. Besides, the sensitivity analyses become easy and simple to implement by employing the automatic differentiation. In light of the above outstanding features, it is promising to offer a new route without using numerical solvers to handle complex issues in structural optimization.

Table 22 Optimization results obtained for the 120-bar dome truss

A_i (in^2)	Kaveh [17]	Kaveh [55]	Kaveh [56]	Talatahari [16]	Kaveh [18]	Kaveh [15]	Adil [54]	This study	
								DE	PINEFN
A_1	3.095	3.027	3.0244	3.0244	3.0241	3.0242	3.0241	1.9124	1.9171
A_2	14.405	14.606	14.7168	14.7804	14.7809	14.6893	14.7826	14.8834	14.9284
A_3	5.02	5.044	5.08	5.0567	5.0522	5.0882	5.0512	5.7567	5.7528
A_4	3.352	3.139	3.1374	3.1359	3.1369	3.1392	3.137	2.5652	2.5667
A_5	8.631	8.543	8.5012	8.483	8.5004	8.5164	8.4981	10.0474	10.0309
A_6	3.432	3.367	3.3019	3.3104	3.2888	3.2857	3.2916	3.5773	3.5785
A_7	2.499	2.497	2.4965	2.4977	2.4969	2.4964	2.4968	1.9795	1.9759
W_{best} (lb)	33,248.90	33,251.90	33,250.42	33,251.22	33,250.05	33,250.01	33,249.75	32,499.65	32,505.93
CVE_{max}	–	–	–	–	–	–	–	None	None
Times (s)	–	–	–	–	–	–	–	2311.08	20.89

Table 23 Error of the constraints for the 120-bar dome truss

Elements	Stress (ksi)			Dofs	Displacement (in)		
	FEA	PINEFN	Error (%)		FEA	PINEFN	Error (%)
1	− 2.0933	− 2.0933	0.0011	w_1	− 0.19689	− 0.19690	0.0003
13	− 2.2703	− 2.2703	0.0002	w_2	− 0.19685	− 0.19686	0.0007
25	− 2.2676	− 2.2676	0.0002	w_{14}	0.01774	0.01774	0.0038
37	− 2.2043	− 2.2044	0.0018	w_{15}	− 0.02062	− 0.02062	0.0068
61	1.83995	1.8400	0.0005				
85	− 2.7892	− 2.7892	0.0003				
97	− 2.8292	− 2.8292	0.0025				
Max	1.8399	1.8400	0.0025	–	0.0177	0.0177	0.0068
Min	− 2.8292	− 2.8292	0.0002	–	− 0.1969	− 0.1969	0.0003

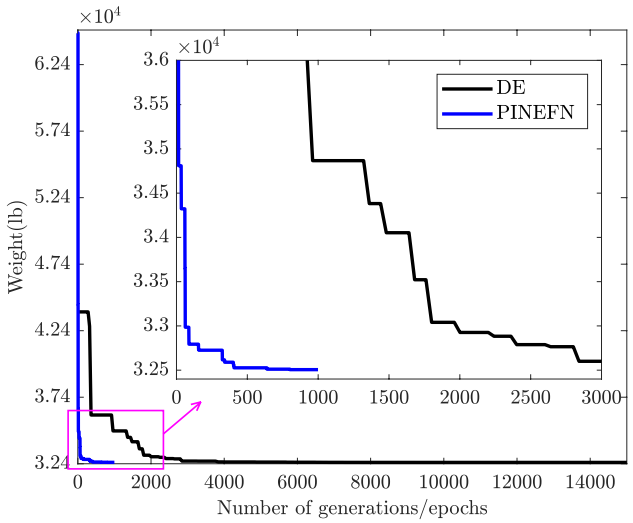

Fig. 17 The weight convergence histories of the 12-bar dome truss obtained using the PINEFN and DE

Acknowledgements This research was supported by a grant (NRF-2021R1A2B5B03002410) from NRF (National Research Foundation of Korea) funded by MEST (Ministry of Education and Science Technology) of Korean government.

Author contributions HMT: conceptualization, methodology, software, formal analysis, investigation, writing - original draft, writing review & editing, visualization. DDM: writing - original draft, writing review & editing. JK: data curation, validation. JaewL: data curation, validation. JaehL: vonceptualization, methodology, supervision, funding acquisition.

Declaration

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

References

1. Khot N (1983) Nonlinear analysis of optimized structure with constraints on systemstability. AIAA J 21:1181–1186
2. Khot N, Berke L (1984) Structural optimization using optimality criteria methods
3. Khot N, Kamat M (1985) Minimum weight design of truss structures with geometric nonlinear behavior. AIAA J 23:139–144

4. Rizzi P (1976) Optimization of multi-constrained structures based on optimality criteria? In: 17th structures, structural dynamics, and materials conference, p 1547
5. Hrinda GA, Nguyen DT (2008) Optimization of stability-constrained geometrically nonlinear shallow trusses using an arc length sparse method with a strain energy density approach. Finite Elements in Analysis and Design 44:933–950
6. Saka M, Ulker M (1992) Optimum design of geometrically nonlinear space trusses. Comput Struct 42:289–299
7. Schmit L Jr, Farshi B (1974) Some approximation concepts for structural synthesis. AIAA J 12:692–699
8. Lieu QX (2022) A novel topology framework for simultaneous topology, size and shape optimization of trusses under static, free vibration and transient behavior, Engineering with Computers, pp 1–25
9. Lieu QX, Do DT, Lee J (2018) An adaptive hybrid evolutionary firefly algorithm for shape and size optimization of truss structures with frequency constraints. Comput Struct 195:99–112
10. Lee KS, Geem ZW (2004) A new structural optimization method based on the harmony search algorithm. Comput Struct 82:781–798
11. Degertekin S (2012) Improved harmony search algorithms for sizing optimization of truss structures. Comput Struct 92:229–241
12. Adeli H, Kumar S (1995) Distributed genetic algorithm for structural optimization. J Aerospace Eng 8:156–163
13. Sarma KC, Adeli H (2000) Fuzzy genetic algorithm for optimization of steel structures. J Struct Eng 126:596–604
14. Li L-J, Huang Z, Liu F, Wu Q (2007) A heuristic particle swarm optimizer for optimization of pin connected structures. Comput Struct 85:340–349
15. Kaveh A, Ghazaan MI (2017) Optimum design of skeletal structures using pso-based algorithms, Periodica Polytechnica. Civil Eng 61:184–195
16. Talatahari S, Kheirollahi M, Farahmandpour C, Gandomi AH (2013) A multi-stage particle swarm for optimum design of truss structures. Neural Comput Appl 23:1297–1309
17. Kaveh A, Talatahari S (2009) Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures. Comput Struct 87:267–283
18. Kaveh A, Bakhshpoori T, Afshari E (2014) An efficient hybrid particle swarm and swallow swarm optimization algorithm. Comput Struct 143:40–59
19. Pierezan J, dos Santos Coelho L, Mariani VC, de Vasconcelos Segundo EH, Prayogo D (2021) Chaotic coyote algorithm applied to truss optimization problems. Comput Struct 242:106353
20. Kaveh A, Talatahari S (2009) Size optimization of space trusses using big bang-big crunch algorithm. Comput Struct 87:1129–1140
21. Camp CV (2007) Design of space trusses using big bang-big crunch optimization. J Struct Eng 133:999–1008
22. Degertekin S, Hayalioglu M (2013) Sizing truss structures using teaching-learning-based optimization. Comput Struct 119:177–188
23. Dang KD, Nguyen-Van S, Thai S, Lee S, Luong VH, Lieu QX (2022) A single step optimization method for topology, size and shape of trusses using hybrid differential evolution and symbiotic organisms search. Comput Struct 270:106846
24. Nguyen-Van S, Nguyen KT, Dang KD, Nguyen NT, Lee S, Lieu QX (2021) An evolutionary symbiotic organisms search for multiconstraint truss optimization under free vibration and transient behavior. Adv Eng Softw 160:103045
25. Mai HT, Lieu QX, Kang J, Lee J (2022) A robust unsupervised neural network framework for geometrically nonlinear analysis of inelastic truss structures. Appl Math Model 107:332–352
26. He Y, Zhang L, Chen Z, Li CY (2022) A framework of structural damage detection for civil structures using a combined multi-scale convolutional neural network and echo state network. Eng Comput. https://doi.org/10.1007/s00366-021-01584-4
27. Lieu QX, Nguyen KT, Dang KD, Lee S, Kang J, Lee J (2022) An adaptive surrogate model to structural reliability analysis using deep neural network. Expert Syst Appl 189:116104
28. Mai HT, Lieu QX, Kang J, Lee J (2022) A novel deep unsupervised learning-based framework for optimization of truss structures. Eng Comput. https://doi.org/10.1007/s00366-022-01636-3
29. Hajela P, Berke L (1991) Neurobiological computational models in structural analysis and design. Comput Struct 41:657–667
30. Hajela P, Berke L (1991) Neural network based decomposition in optimal structural synthesis. Comput Syst Eng 2:473–481
31. Adeli H, Park HS (1995) Optimization of space structures by neural dynamics. Neural Netw 8:769–781
32. Ramasamy J, Rajasekaran S (1996) Artificial neural network and genetic algorithm for the design optimizaton of industrial roofs-a comparison. Comput Struct 58:747–755
33. Mai HT, Kang J, Lee J (2021) A machine learning-based surrogate model for optimization of truss structures with geometrically nonlinear behavior. Finite Elements in Analysis and Design 196:103572
34. Li B, Huang C, Li X, Zheng S, Hong J (2019) Non-iterative structural topology optimization using deep learning. Computer-Aided Design 115:172–180
35. White DA, Arrighi WJ, Kudo J, Watts SE (2019) Multiscale topology optimization using neural network surrogate models. Comput Methods Appl Mech Eng 346:1118–1135
36. Chi H, Zhang Y, Tang TLE, Mirabella L, Dalloro L, Song L, Paulino GH (2021) Universal machine learning for topology optimization. Comput Methods Appl Mech Eng 375:112739
37. Chandrasekhar A, Suresh K (2021) Tounn: topology optimization using neural networks. Struct Multidiscip Optimiz 63:1135–1149
38. Chandrasekhar A, Sridhara S, Suresh K (2022a) Gm-tounn: Graded multiscale topology optimization using neural networks, arXiv preprint arXiv:2204.06682
39. Chandrasekhar A, Mirzendehdel A, Behandish M, Suresh K (2022b) Frc-tounn: Topology optimization of continuous fiber reinforced composites using neural network, arXiv preprint arXiv:2205.03737
40. Li W, Bazant MZ, Zhu J (2021) A physics-guided neural network framework for elastic plates: comparison of governing equations-based and energy-based approaches. Comput Methods Appl Mech Eng 383:113933
41. Raissi M, Perdikaris P, Karniadakis GE (2019) Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys 378:686–707
42. Ohkubo S, Watada Y, Toshio F (1987) Nonlinear analysis of truss by energy minimization. Comput Struct 27:129–145
43. Tran HC, Lee J (2013) Force methods for trusses with elastic boundary conditions. Int J Mech Sci 66:202–213
44. Rahami H, Kaveh A, Gholipour Y (2008) Sizing, geometry and topology optimization of trusses via force method and genetic algorithm. Eng Struct 30:2360–2369
45. Sonmez M (2011) Artificial bee colony algorithm for optimization of truss structures. Appl Soft Comput 11:2406–2418
46. Hasançebi O (2008) Adaptive evolution strategies in structural optimization: enhancing their computational performance with applications to large-scale structures. Comput Struct 86:119–132
47. Kingma DP, Ba J (2014) Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980
48. Manning C, Socher R (2017) Natural language processing with deep learning, Lecture Notes Stanford University School of Engineering

49. Lamberti L (2008) An efficient simulated annealing algorithm for design optimization of truss structures. Comput Struct 86:1936–1953
50. Bekdaş G, Nigdeli SM, Yang X-S (2015) Sizing optimization of truss structures using flower pollination algorithm. Appl Soft Comput 37:322–331
51. Pouriyanezhad E, Rahami H, Mirhosseini S (2021) Truss optimization using eigenvectors of the covariance matrix. Eng Comput 37:2207–2224
52. Adeli H, Kamal O (1986) Efficient optimization of space trusses. Comput Struct 24:501–511
53. Construction A (1989) Manual of steel construction: allowable stress design. American Institute of Steel Construction, Chicago, IL, USA
54. Adil B, Cengiz B (2020) Optimal design of truss structures using weighted superposition attraction algorithm. Eng Comput 36:965–979
55. Kaveh A, Talatahari S (2010) Optimal design of skeletal structures via the charged system search algorithm. Struct Multidiscip Optimiz 41:893–911
56. Kaveh A, Bakhshpoori T (2013) Optimum design of space trusses using cuckoo search algorithm with levy flights. Iran J Sci Technol - Trans Civil Eng 37(C1):1–15

Authors and Affiliations

Hau T. Mai[1,2] · Dai D. Mai[3] · Joowon Kang[4] · Jaewook Lee[1] · Jaehong Lee[1]

Hau T. Mai
maitienhaunx@gmail.com; maitienhau@iuh.edu.vn

Dai D. Mai
daimd@hcmute.edu.vn

Joowon Kang
kangj@ynu.ac.kr

Jaewook Lee
jaewook@sejong.ac.kr

[1] Deep Learning Architectural Research Center, Sejong University, 209 Neungdong-ro, Gwangjin-gu, Seoul 05006, Republic of Korea

[2] Faculty of Mechanical Technology, Industrial University of Ho Chi Minh City, Ho Chi Minh City, Vietnam

[3] Faculty of Mechanical Engineering, Ho Chi Minh City University of Technology and Education, Ho Chi Minh City, Vietnam

[4] School of Architecture, Yeungnam University, 280, Daehak-Ro, Gyeongsan, Gyeongbuk 38541, Republic of Korea