Advanced Steel Construction – Vol. 22 No. 1 (2026) 105–115
DOI:10.18057/IJASC.2026.22.1.9

EFFICIENT PRESTRESS OPTIMIZATION OF SUSPEN-DOME USING NSGA-III AND MACHINE LEARNING-BASED SURROGATE MODELS

Jin Wang, Ming-Liang Zhu * and Ze-Yun Jin

School of Civil Engineering, Southeast University, Nanjing 210096, China
** (Corresponding author: E-mail: zhumingliang@seu.edu.cn)*

ABSTRACT

Prestress optimization is a critical step in the structural design of suspen-dome, often requiring extensive and time-consuming iterative computations. This study proposes a hybrid framework that integrates the NSGA-III algorithm with machine learning-based surrogate models to address the prestress multi-objective optimization problem of suspen-dome. A comparative analysis of three machine learning algorithms—Deep Belief Network (DBN), Sequence-to-Sequence (Seq2Seq), and Backpropagation Neural Network (BPNN)—is conducted to evaluate surrogate modeling performance. The multi-objective optimization considers four objective functions, and the NSGA-III algorithm is employed to effectively obtain the Pareto front. The optimal solution is selected using multi-criteria decision-making, and a case study is presented to validate the accuracy and efficiency of the proposed method. Results show that the BPNN-based surrogate-assisted optimization achieves the best overall efficiency. The introduction of surrogate models reduces computation time by 95% while maintaining optimization performance comparable to traditional finite element analysis (FEA)-based methods.

ARTICLE HISTORY

Received: 9 December 2024
Revised: 20 June 2025
Accepted: 20 June 2025

KEYWORDS

Suspen-dome;
Prestress optimization;
Multi-objective;
Optimal solution;
Finite element analysis (FEA)

1. Introduction

The suspen-dome, as recommended by Kawaguchi et al.[1], is a hybrid spatial structure consisting of an upper single-layer shell and a lower cable-strut system. The suspen-dome structure is extensively applied in large-span projects, including stadiums and airports, due to its high structural strength and low self-weight[2-4].

After applying prestress, the lower cable-strut system can significantly improve the integral structural performance, enhancing the structure's stability and load-carrying capacity [5]. For the cable-strut system, optimization is usually carried out in terms of minimizing construction costs, reducing self-weight, and optimizing stiffness[6,7]. However, how to design the prestress to achieve these optimization objectives simultaneously remains a challenging question, and this problem is currently addressed mainly through optimization algorithms and finite element iterations. Kitipornchai et al.[8], through the prestress design, systematically considered geometric imperfections, asymmetric loads, span ratios, and connection stiffness to study the buckling problem of the structure. Zhang et al.[9] studied the effect of prestress on the maximum displacement and critical load-carrying capacity of the structure through numerical simulation of the cable in the lattice shell structure but did not optimize the prestress. Kang et al.[10] devised a particular annular suspen-dome structure . The prestress of different loop cables is optimized using a genetic algorithm, enhancing the overall stability of the structure. However, this approach is only suitable for relatively simple structures. Liang et al.[11] recommended a structural prestress optimal model, converting multi-objective optimization into single-objective optimization, thus addressing the collaborative optimization of multiple objectives, but this method is highly subjective. Kaveh[12] used the ECBO algorithm to settle various optimization problems related to the geometry, size, and prestress of suspen-dome, significantly improving structural performance and economy, although the computational efficiency was relatively low. Olofin[13] replaced the steel cables of the suspen-dome structure with carbon fiber-reinforced polymer (CFRP) for prestress design, which proved to be highly effective.

In the past few decades, the handling of multi-objective optimization problems has become increasingly important, especially the issue of balancing different objectives[14,15]. Ishibuchi[16] studied multi-objective optimization algorithms and used evolutionary-based optimization algorithms to solve multi-objective optimization problems, achieving a dynamic balance among multiple objectives. Ma et al.[17] recommended a multi-objective optimization method for the prestress optimization of cable-strut structures, balancing cross-sectional area, stiffness matrix eigenvalues, prestress variance, and prestress error. However, this method requires significant computational power. Chen et al.[18] successfully optimized feasible prestress patterns of cable-strut structures using a multi-objective particle swarm algorithm, greatly improving efficiency, though it mainly applies to simple structures. In multi-objective optimization, the Pareto optimal solution set represents a collection of non-dominated solutions that achieve a balance among multiple objectives. Whereas, in company with the dimensionality of optimization problems increases, frequently used algorithms like NSGA-II[19] and SPEA2[20] encounter multiple challenges[21,22], and most algorithms are difficult to deal with four or more objectives effectively. To tackle these challenges, researchers have recommended upgraded Pareto-based optimization methods and new algorithm designs[23], which help to better explore the Pareto frontier[24]. This guarantees decision-makers to more comprehensively assess solutions multiplicity and scope, permitting them to opt for the optimal solution that best meets their demands [25]. NSGA-III[15] improves the multiplicity and uniformity of solutions by introducing evenly distributed reference points, especially effective in high-dimensional spaces, reinforcing the scope of the Pareto front and maintaining a uniform allocation within solutions.

With the advancement of computer technology applications, machine learning has also profoundly impacted traditional engineering fields[26]. Its powerful predictive capabilities have been widely applied in fault prediction for structural health monitoring[27] and material property prediction[28]. In the design field, machine learning is used for automated generation[29] and optimization of design schemes[30]. Additionally, real-time data learning is employed to adjust control strategies[31]. Furthermore, optimization based on machine learning surrogate models has been widely applied, significantly improving optimization efficiency. Li[32] proposed a surrogate model optimization algorithm that combines the low computational cost of surrogate models with the rapid convergence of Bayesian algorithms, showing strong applicability. Zhou[33] utilized machine learning algorithms to establish a substitute model and perform optimization, greatly enhancing optimization efficiency.

This research proposes a multi-objective prestress optimization framework by integrating NSGA-III with machine learning. The remainder of the paper is organized as follows: Section 2 outlines the multi-objective prestress optimization problem. In *section* 3, a brief description of NSGA-III optimization algorithm and three surrogate models based on machine learning is given. *Section* 4 conducts a case analysis of two different configurations of the suspen-dome, and *Section* 5 provides a summary of the conclusions.

2. Framework

2.1. Objective function

2.1.1. Vertical displacement

The prestress of the suspen-dome structure directly affects the structural stiffness and the selection of cross-sections. Under normal operating conditions, the optimal prestress state should keep the structural stiffness within an appropriate range. Excessive prestress can lead to overly high stiffness, resulting in material waste, while insufficient prestress can increase deflection, affecting the structure's normal performance. Therefore, the maximum vertical

displacement is generally used to measure the structural stiffness, and the objective function is expressed as follows:

$$f_1 = min(|u_z|_{max}) \quad (1)$$

where u_z denotes the maximum nodal vertical displacement.

2.1.2. Horizontal displacement

The lower cable-strut system of the suspen-dome can balance the support reactions of the upper single-layer lattice shell. To reduce construction costs, the radial constraints of the structure are usually released. Under the action of prestress, radial displacement will occur at the structural supports. The radial displacement at the supports of the suspen-dome is set as the optimization objective function, and its expression is as follows:

$$f_2 = min\,(|u_x|_{max}) \quad (2)$$

here, u_x denotes the maximum displacement of the support node.

2.1.3. Steel quantity

For the structural system, minimizing the amount of steel used can reduce the structure's self-weight, thereby enhancing overall performance and economy. By reasonably controlling the amount of steel, it is possible to achieve optimal economic benefits while meeting the comprehensive requirements of structural load, material strength, and design constraints. Therefore, optimizing the steel usage in the lower cable-strut system of the suspen-dome is an effective strategy. The objective function is expressed as follows:

$$f_3 = min(\sum_{i=1}^{n} \rho A_i L_i) \quad (3)$$

where $\boldsymbol{\rho}$ denotes the material density, A_i represents the cross-sectional area, and L_i represents the element length.

2.1.4. Strain energy

The minimum strain energy of the suspen-dome is crucial in design. By minimizing strain energy, the integral energy consumption of the structure can also be significantly reduced, thereby lowering energy loss during actual use and achieving a more sustainable architectural design. The objective function is expressed as follows:

$$f_4 = min\,(E) \quad (4)$$

where E is the total strain energy of the structure.

2.2. Constraint conditions

Based on the above optimization process, there may be solution sets that do not meet the constraints. Therefore, it is necessary to establish constraint conditions, assigning a lower rank to solutions that fail to satisfy these constraints. By setting appropriate constraints, a solution set that meets the specified conditions can be obtained.

2.2.1. Displacement limit

Displacement limit conditions are key restrictions to undertake the safety and normal operation of the structure. By setting the maximum allowable displacement of the structure or its components under extreme conditions, damage can be prevented, and the stability and reliability of the structure can be guaranteed. According to reference [34], the specified limits are as follows:

$$u_{max} \leq H/250 \quad (5)$$

where u_{max} denotes the maximum deflection value.

2.2.2. Stress condition

The struts supporting the upper dome structure are under compression, with the maximum compressive force not exceeding the ultimate load-bearing capacity. Additionally, a partial factor of 0.5 is applied to the cables, meaning the cable force should not exceed 50% of the cable's ultimate load-bearing capacity [34]. Additionally, it must be ensured that the cables do not experience slack, i.e., the cable force must be greater than zero. The expressions are as follows:

$$\begin{cases} 0 < \sigma_t < [\sigma_t] \\ -[\sigma_c] \leq \sigma_c \leq 0 \end{cases} \quad (6)$$

where σ_t and σ_c represent the stresses in the cable-strut structure, and $[\sigma_t]$ and $[\sigma_c]$ represent their ultimate stresses.

2.2.3. Strut stability

As compression members, the struts must ensure their stability under axial pressure. By introducing the coefficient φ, the stability of the struts can be effectively guaranteed, preventing buckling. The expression is as follows:

$$\varphi = \frac{N}{A \times f} \quad (7)$$

where φ is the stability coefficient, N is the axial force, A is the cross-sectional area, and f is the design compressive member's strength.

2.3. Multi-criteria decision analysis

By performing multi-objective optimization on the above four objectives, the Pareto front can be obtained. The Pareto front consists of a set of competing solutions, where each solution outperforms others in certain objectives but may perform worse in others. Since the solutions on the Pareto front cannot be directly compared across multiple dimensions, a method is needed to comprehensively consider the importance of different objectives and select the optimal solution. To this end, this paper adopts the Analytic Hierarchy Process (AHP).

Step1: Decompose the decision-making problem into three levels: the goal level, the criteria level, and the alternatives level. These correspond to the final objective of the decision, the standard influencing the decision, and the alternative options for decision-making, respectively.

Step2: According to the degree of importance, use a scale of 1 to 9, where a higher number indicates greater importance; use this to construct the judgment matrix.

Step3: Based on the judgment matrix, use the eigenvector method to compute the relative weight of each criterion.

Step4: The Consistency Ratio (CR) is used to ensure the logical consistency of the judgment matrix in pairwise comparisons, thereby improving the reliability of weight assignments. The steps for calculating CR are as follows: first, the largest eigenvalue (λ_{max}) of the judgment matrix is determined through eigenvalue decomposition; then, the CR value is calculated using formula. If the CR value is less than 0.1, the consistency of the judgment matrix is considered acceptable, indicating reasonable weight assignments; if the CR value is greater than 0.1, the judgment matrix needs to be adjusted. Calculating CR ensures the scientific rigor and consistency of AHP in multi-criteria decision-making.

$$CR = \frac{\lambda_{max} - n}{(n-1) \times RI} \quad (8)$$

Step5: For each solution, calculate the overall score based on the product of weights for all criteria, as follows:

$$S_j = \sum_{i=1}^{m} \omega_i a_{ij} \quad (9)$$

where w_i is the weight of the i-*th* criterion, and a_{ij} is the score for that criterion.

Step6: Select the solution with the highest score as the final decision.

3. Prestress optimization framework based on surrogate model

The use of external finite element analysis includes both online and offline approaches. In the former, optimization iterations directly call the external finite element program for calculations and return the analysis results to the optimizer. This approach offers high accuracy but incurs significant computational costs, potentially resulting in a lengthy optimization process, especially as each iteration requires considerable time to run the external finite element solver.

In offline analysis, multiple finite element analyses are conducted in advance, with the results stored in a database or neural network surrogate model before optimization begins. During optimization, this database or model is accessed to predict responses under new conditions, eliminating the need for real-time finite element calculations. This method is computationally efficient, though accuracy may be limited by the quantity and scope of prior analyses. To simplify computations, this study has trained a highly accurate neural network surrogate model to streamline analysis. By inputting the structure's prestress, the model outputs internal forces and displacements to predict structural responses. Additionally, a multi-objective algorithm is introduced into the surrogate model to optimize the prestress of the cable-dome structure.

3.1. Machine learning

3.1.1. DBN

Deep Belief Networks (DBN) are a sort of deep learning model composed of numerous layers of Restricted Boltzmann Machines (RBM). They were recommended by Hinton et al.[35] in 2006. DBNs can effectively learn deep feature representations of data through unsupervised, layer-by-layer pre-training. These features can then be utilized for assignments such as classification and regression, allowing DBNs to reveal hidden, hierarchical structural information within the data.

Fig. 1 and Fig. 2 illustrate the core structure of a DBN, which includes a visible layer and multiple hidden layers. Each pair of layers is connected by an RBM. The visible layer is responsible for receiving input datum, whereas each hidden layer processes more abstract representations of data. Multiple RBM layers followed by one or more backpropagation layers are used for fine-tuning. During the training of a DBN, each RBM layer is trained sequentially, utilizing the output from the prior layer as its input to learn higher-level features of the data. After training is complete, the DBN is fine-tuned through added supervised learning layers to enhance performance for specific tasks. In the final layer of the DBN, a BP (Back Propagation) network is set up to accept the output characteristic vectors from the RBMs as its input feature vectors and is trained in a supervised manner to form the neural network.

$$E(E(V,h) = -\sum_i a_i V_i - \sum_j b_j H_j - \sum_{i,j} V_i W_{ij} H_j) \tag{10}$$

In this context, V and H represent the nodes in the visible and hidden layers, respectively, a_i and b_j are the bias terms, and W_{ij} represents the weights between nodes. The training of DBNs involves using this energy function for layer-by-layer pre-training, followed by possible fine-tuning during a subsequent supervised learning phase. Thus, DBNs can capture complex features and patterns in data, possessing powerful predictive and classification capabilities, especially in scenarios involving large volumes of unlabeled data. As technology advances, the potential of DBNs in handling complex data analysis tasks is continually being explored.

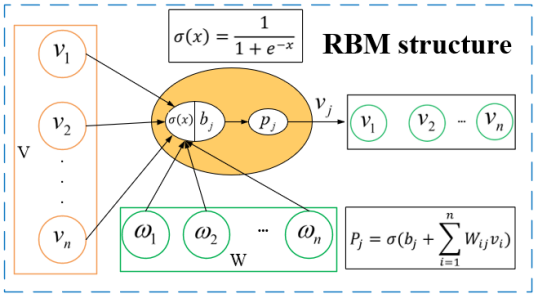


Fig. 1 Proposed architecture of the restricted boltzmann machine (RBM)

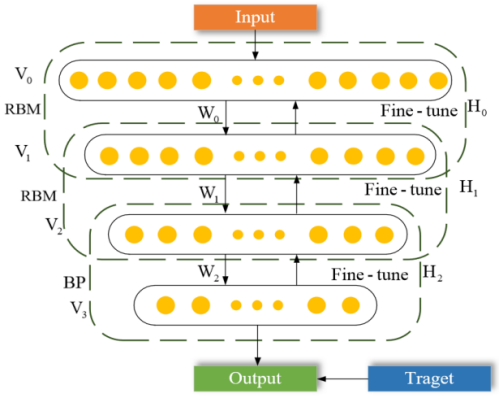


Fig. 2 Hierarchical flowchart of the deep belief network (DBN) training process

3.1.2. Seq2Seq

The Seq2Seq model was introduced in 2015 by Vinyals and Le[36]. The design concept involves two RNN (Recurrent Neural Network) components, an encoder and a decoder, to handle varying input and output sequences. The encoder is accountable for transforming the input sequence into a fixed-length context vector, while the decoder yields the target sequence based on this vector. The core architecture is shown in Fig.3.

Encoder: $h_t = f(x_t, h_{t-1})$, where h_t is the hidden state at time step$t$, and x_t is an element of the input data.

Context Vector: $c = q(\{h_1, h_2, \ldots, h_T\})$, where T is the length of the input data, and the function q is usually taking the last hidden state.

Decoder: Given the context vector c and all previous outputs $y_1, \ldots, y_{t-1}$, it generates the next output y_t. The initial input is set as $s_0 = c$ and the sequence is generated step by step according to the following recurrence:

$$s_t = f(s_{t-1}, y_{t-1}),\ y_t = softmax(W_0 s_t + b_0) \tag{11}$$

Encoder: In the encoder, each input symbol x_t is embedded into a vector $e_t = Embed(x_t) \in R^d$, and passed into a recurrent structure to compute the hidden state recursively:

$$h_t = \Phi(W_x e_t + W_h h_{t-} + b_h),\ \ t = 1,2,\cdots,T \tag{12}$$

where $\phi(\cdot)$is a nonlinear activation function, W_x, W_h are weight matrices, and b_h is a bias term. The final hidden state h_T is regarded as the compressed representation of the entire input sequence, namely the context vector $c = h_T$.

The decoder begins from the initial state $s_0 = c$, and at each step, it takes the output from the previous time step as input to generate the current hidden state and output probability distribution. Specifically, the input symbol at the t-th decoding step is y_{t-1}, which is embedded as $et' = Embed(y_{t-1})$, The hidden state is then updated as:

$$s_t = \Phi\left(W_y e_t^{'} + W_s s_{t-} + b_s\right) \tag{13}$$

Subsequently, the model computes the predicted probability distribution for the current step using a linear transformation followed by a *Softmax* function:

$$\tilde{y}_t = Softmax(W_o s_t + b_o) \tag{14}$$

where $y^t \in R^V$, represents the probability distribution over all words in the target vocabulary, and V is the vocabulary size.

The output sequence is generated in an autoregressive manner. During training, the ground-truth token y_{t-1} (rather than the model's prediction) is fed as input to the next time step (teacher forcing). The model is trained to maximize the conditional probability of the entire target sequence, which corresponds to minimizing the negative log-likelihood loss:

$$L = -\sum_{t=1}^{T'} LogP(y_t | y_1, \cdots, y_{t-1}, x) \tag{15}$$

The conditional probability at each step is determined by the *Softmax* output:

$$P(y_t | y_{<t}, \boldsymbol{x}) = \frac{exp(W_{y_t} s_t)}{\sum_j^V exp(w_j^T s_t)} \tag{16}$$

The Seq2Seq model, through its encoder-decoder structure, models the conditional mapping from variable-length input sequences to variable-length output sequences. It constitutes a sequence of stateful nonlinear transformations and nested probabilistic prediction processes.

The Seq2Seq model is highly flexible and can handle input and output of any length, making it adaptable to various data types. However, it faces challenges such as a heavy burden on the decoder, which must recover all output information from a single fixed-length vector. Additionally, training the model is complex, especially prone to issues like gradient vanishing and explosion in long sequences. Moreover, its high computational demands can limit its use in resource-constrained environments.

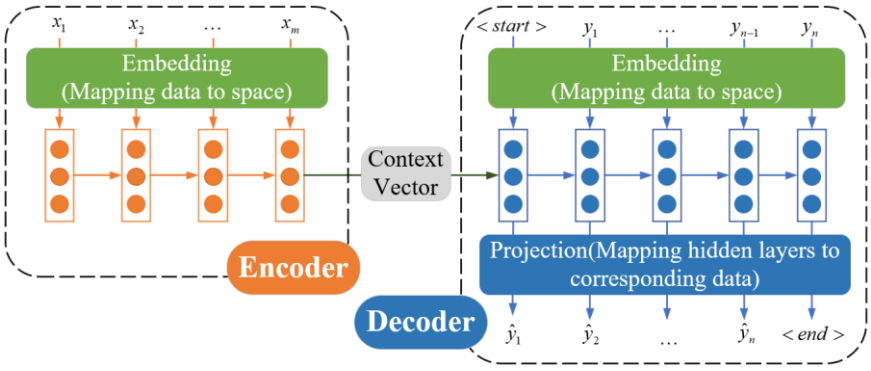


Fig. 3 Illustration of the encoder-decoder architecture in Seq2Seq modeling

3.1.3. BPNN

The Backpropagation Neural Network (BPNN) is a sort of multilayer feed-forward neural network introduced in 1986 by Rumelhart et al.[37]. The training of a BPNN is conducted using the error backpropagation algorithm. This process primarily involves the optimization of a nonlinear objective function. The BP network employs numerical optimization methods to adjust weights, utilizing not only the first-order derivative information of the objective function but frequently also the second-order derivative information.

The structure of the BPNN is shown in Fig.4 and can be uniformly described as:

$$f\left(\boldsymbol{x}^{(k+1)}\right) = \min f\left(\boldsymbol{x}^{(k)} + \omega^{(k)} S(\boldsymbol{x}^{(k)})\right) \tag{17}$$

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)(\eta)} + \omega^{(k)} S\left(\boldsymbol{x}^{(k)}\right) \tag{18}$$

where $\boldsymbol{x}^{(k+1)}$ represents the vector consisting of all weights and biases of the network; $S(\boldsymbol{x}^{(k)})$ is the search direction in the vector space formed by the components of x; $\eta^{(k)}$ is the step length in the direction of $S(\boldsymbol{x}^{(k)})$ that minimizes $f\left(\boldsymbol{x}^{(k+1)}\right)$.

Thus, the optimization of network weights can be divided into two steps: first, determine the best search direction $S(\boldsymbol{x}^{(k)})$ for the current iteration, and then seek the most optimal iteration step length in this direction.

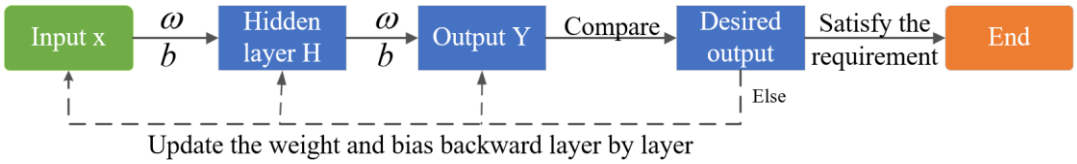


Fig. 4 BPNN Training flowchart

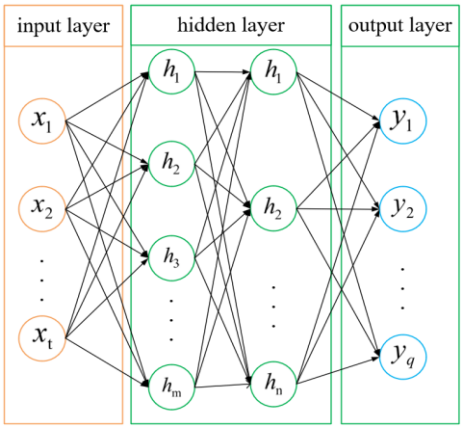


Fig. 5 Architectural diagram of the backpropagation neural network (BPNN)

Fig. 5 introduces the general architecture of BPNN, BPNN is an artificial neural network model used for supervised learning. It calculates output values through forward propagation and adjusts the weights and biases in the network using a backpropagation algorithm, gradually bringing predictions closer to actual values. The structure of BPNN includes an input layer, one or more hidden layers, and an output layer. The input layer receives feature data, the hidden layer transforms features through neurons and activation functions, and the output layer generates the prediction results. In the forward propagation phase, input data flows from the input layer to the output layer, with each neuron computing a weighted sum and generating output signals through an activation function. In the backpropagation phase, the error between the output value and the target value is calculated, and this error is propagated back layer by layer from the output layer to adjust the weights and biases in each layer, gradually reducing the error until the network converges.

$$\alpha_h = \Sigma_{i=1}^{d} v_{ih} x_i + \theta_h \ (i = 1,2,\ldots,d; h = 1,2,\ldots,q) \tag{19}$$

From the hidden layer to the output layer:

$$\beta_j = \Sigma_{h=1}^{q} w_{hj} b_h + \theta_j \ (j = 1,2,\ldots,l; h = 1,2,\ldots,q) \tag{20}$$

Back propagation is to modify the network parameters by computing the error between the output layer and the expected value, so as to the fault gets smaller. The calculation error formula is as follows:

$$E = \frac{1}{2}\Sigma_{k=1}^{l}(y_k - T_k)^2 \tag{21}$$

3.2. Multi-objective optimization algorithm

Although traditional prestress optimization algorithms based on finite element analysis are widely used, they consume considerable time and resources, and they often struggle to balance safety and economic efficiency. As the complexity of prestress optimization design increases, so does the number of objectives that need optimization. In practical engineering, achieving an optimal balance among multiple, often conflicting, objectives is essential. By using multi-objective optimization algorithms, the Pareto frontier and non-dominated solution selection can help identify the best trade-offs among objectives, effectively addressing these challenges. This paper employs three improved multi-objective optimization algorithms to compare their effectiveness in solving the aforementioned multi-objective optimization problems.

NSGA-III (Non-dominated Sorting Genetic Algorithm III) [38,39] is a multi-objective evolutionary algorithm specifically designed to handle complex optimization problems with high-dimensional objective spaces. As illustrated in Fig. 6, NSGA-III builds upon the foundational framework of the NSGA series, with the goal of identifying and maintaining the Pareto front—a set of optimal solutions where no solution is dominated by another across all objectives. The algorithm enhances selection by generating uniformly distributed reference points in the objective space, which helps maintain solution diversity and even distribution. The main steps involve initializing the population, evaluating objective functions, performing non-dominated sorting, and selecting solutions based on reference points. Selected solutions undergo crossover and mutation to create new candidates, updating the population. This process is repeated until the specified number of iterations or convergence conditions are met, resulting in a well-distributed and converged Pareto-optimal set. NSGA-III is widely

applied in fields like engineering and economics, particularly excelling in scenarios requiring the simultaneous optimization of multiple conflicting objectives. Finally, determine if the termination conditions are met. If met, output the final set of non-dominated solutions; otherwise, return to the evaluation step. Through these steps, NSGA-III maintains the diversity and uniformity of solutions while closely approximating the optimal solutions and extensively covering the real Pareto front.

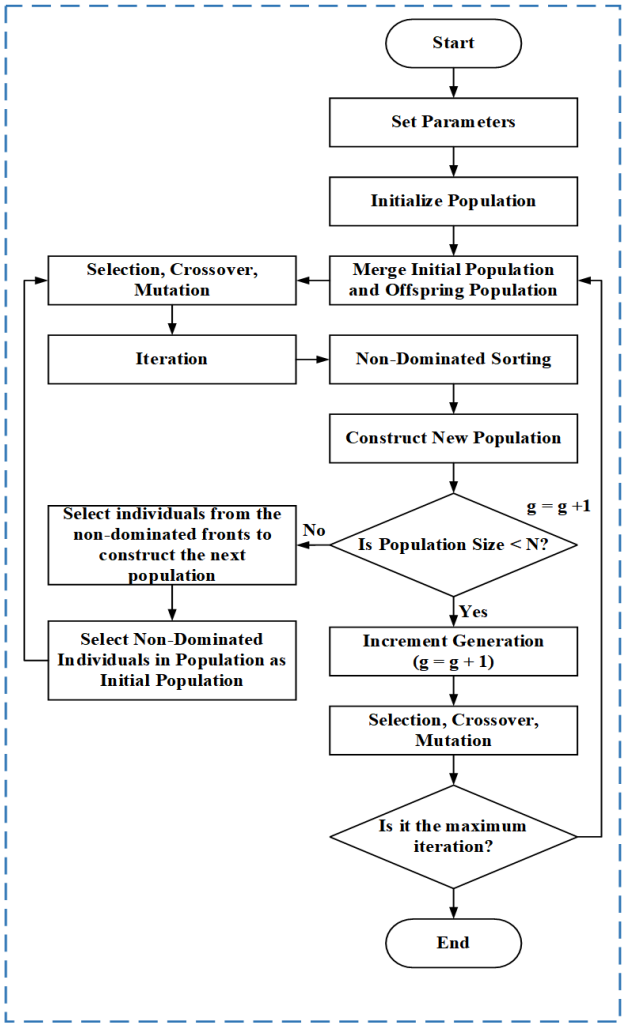


Fig. 6 Process flowchart of the NSGA-III algorithm for multi-objective optimization

3.3. Optimization framework

3.3.1. Step 1: Establish model, group cables and struts, generate initial prestress, and create the dataset

To construct a finite element model of a suspen-dome, the structural symmetry and functional characteristics of the internal cable-struts are utilized to group the rods, reflecting the distribution pattern of internal forces. The prestress range for the cable-struts is determined based on the sectional performance parameters of the cable-strut units. Utilizing the prestress range produced from preliminary design, MATLAB is used to generate initial tension forces uniformly distributed within a specific range. Finite element analysis software then calculates the maximum vertical displacement, lateral support displacement, cable force uniformity coefficient, and maximum cable force value. These calculations form the training dataset for a neural network. This method effectively integrates the structural analysis with machine learning to optimize the design and performance prediction of the suspen-dome.

3.3.2. Step 2: Train the neural network surrogate model

After obtaining the dataset through structural training, the sample dataset is first normalized. Following this, the hyperparameters of the machine learning model are configured to achieve optimal prediction accuracy by setting appropriate model parameters. The dataset is split into training and test sets in a 7:3 ratio. The training set enhances the model's generalization abilities, while the test set evaluates accuracy and stability on new data, establishing a foundation for training the surrogate model. Table 1 shows the parameter settings of the machine learning algorithms applied in this study.

Table 1
Hyperparameters of machine learning algorithms

Algorithms	Hyperparameters
DBN	learning_rate = 0.01, batch_size = 20,number of hidden layers = [30 30],units per layer = 30
Seq2Seq	learning_rate = 0.01, batch_size = 20,number of hidden layers = 20, units per layer = 30
BPNN	learning_rate = 0.01, batch_size = 20,number of hidden layers = 50, units per layer = 30

3.3.3. Step 3: Define the optimization problem and constraints, and carry out the optimization

Define the prestress range for the suspen-dome (Table 2), cross-sectional dimensions, material strengths, and parameters related to the optimization algorithm. Determine the optimization objectives: minimize vertical displacement ($min(f_1)$), minimize horizontal displacement ($min(f_2)$), minimum strain energy ($max(f_3)$), and steel usage of the lower system ($min(f_4)$). Introduce constraints to ensure that the tensile and compressive performance of the components remains within safe limits, and control displacements within the structurally allowable range. Initialize the population, create an initial solution set, and use the surrogate model to represent the initial population or swarm. Select the optimization algorithm, set the optimization parameters, iterate to optimize, and continuously improve the solution set until reaching the maximum iteration count.

3.3.4. Step 4: Select the optimal solution

After obtaining the Pareto frontier, the solution set is normalized. Then, based on the degree of importance, a scale assignment is performed using the AHP. The specific assignment process is as follows:9 for minimum vertical displacement($min(f_1)$),7 for minimum lateral displacement ($min(f_2)$), assigning a weight of 5 to both minimum strain energy ($min(f_3)$) and minimum steel usage ($min(f_4)$) a corresponding judgment matrix is established. Using this matrix, the AHP is performed to compute the final optimal solution on the Pareto front.

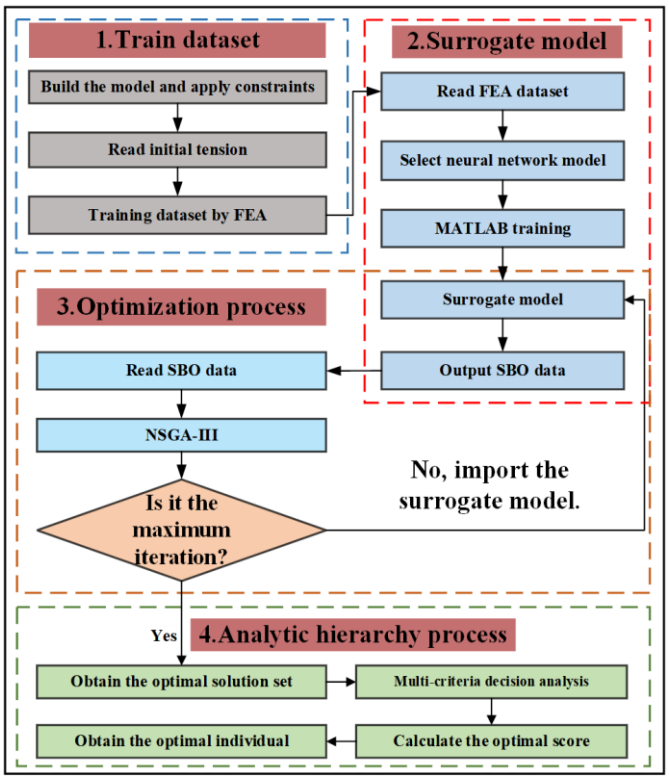
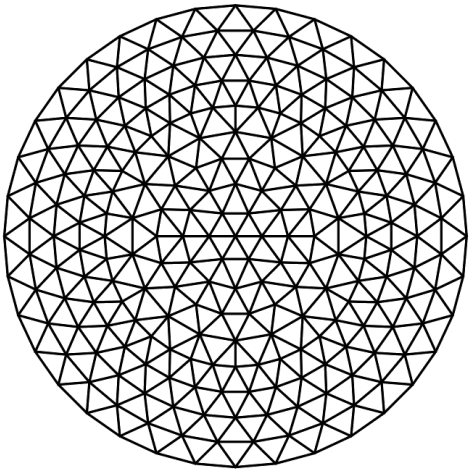


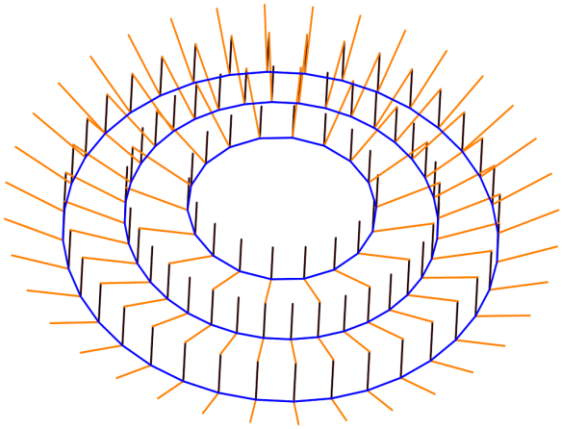
Fig. 7 Flowchart of surrogate-based multi-objective optimization using machine learning algorithms

4. Case study

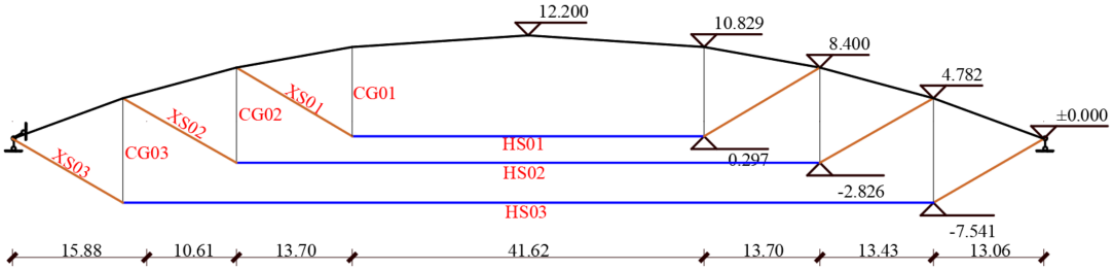
In the previous content, an optimization algorithm based on neural network surrogate models was established through the introduction of three types of neural networks and optimization algorithms. This approach replaces finite element software-based online analysis with offline analysis based on surrogate models. In this section, two different configurations of suspend-dome models are used to verify the effectiveness and accuracy of this method.



(a) Flat view



(b) Top view of the lower structure



(c) Geometrical size(m)

Fig. 8 Analysis model of suspen-dome No. 1

4.1. Case analysis model

4.1.1. Case1 : Suspen-dome1

Fig. 8 presents the axonometric, top, and elevation views of a suspen-dome structure for a stadium. The upper section is a Geiger-type single-layer shell structure, characterized by a 1:10 aspect ratio. The lower structure features a rib-ring arrangement for the cable-strut support system, which includes three circles of cables. The struts in the lower system are made of standard steel with a Young's modulus of 206 GPa, while the cables use high-strength steel with a Young's modulus of 160 GPa. Around the structure, there are hinge constraints in the y and z directions, with radial constraints released. Table 2 shows the range of initial stress inputs used for training the dataset.

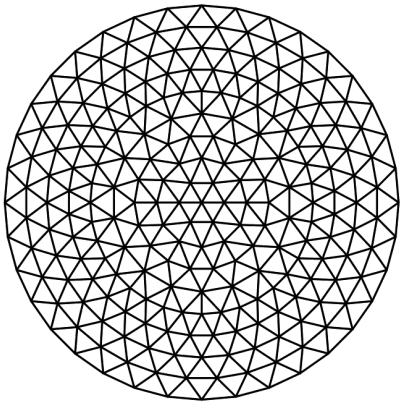
The upper structure of the finite element model is created using the Beam188 element in ANSYS to form a single-layer lattice shell structure. The cable-strut support system in the lower section uses Link180 elements, where the cable elements can only bear tensile forces, and the strut elements can only bear compressive forces. The load conditions are applied according to the requirements of the literature, including a 1.3 times dead load and a 1.5 times live load factor.

4.1.2. Case2 : Levy-type suspen-dome

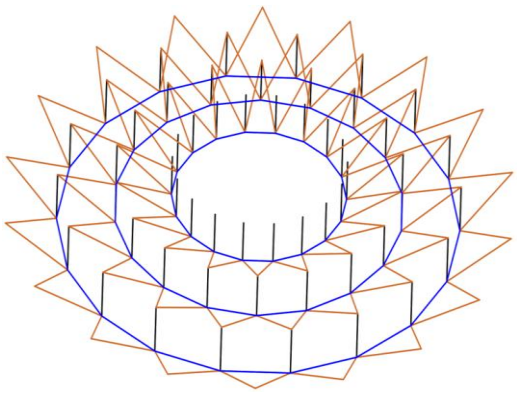
The structure shown in Fig. 9 is a Levy-type suspen-dome, featuring a single-layer shell as the upper structure and a Levy-type cable-strut system as the lower structure, with constraints consistent with those previously described. Due to the structural symmetry, the cable-strut system is categorized into nine groups: three support groups, three diagonal cable groups, and three circular cable groups. According to Table 2, one thousand sets of initial prestress values are randomly generated for data training.

Table 2
Cross-sectional areas of the structural members

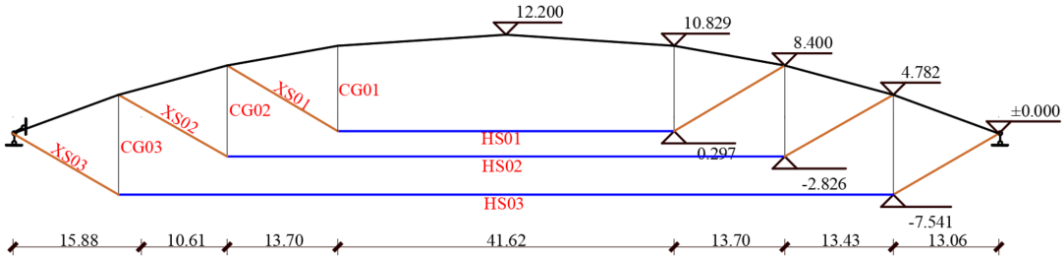
Types	**CG01 ~ 03**			**XS01 ~ 03**			**HS01 ~ 03**		
area/mm^2	8577	5003	4128	6362	2376	1590	9603	3672	1198
min/*k*N	-2000	-1000	-800	600	300	100	800	400	100
max/*k*N	-500	-250	-200	3000	1200	800	4000	2000	500
mean/*k*N	-1250	-625	-500	1800	750	450	2400	1200	300



(a) Flat view



(b) Top view of the lower structure



(c) Geometrical size(m)

Fig. 9 Analysis model of suspen-dome No. 2

4.2. Surrogate model

In Section 3, we explore the application of three distinct machine learning algorithms in building surrogate models to accurately capture the complex relationships between input and output data. The model inputs comprise initial tension forces for nine component groups, which are critical parameters influencing structural performance. Outputs include the internal forces within the ring and diagonal cables and the displacements of nodes, covering a total of 11 parameters to fully characterize the structural response under specified tension conditions. To evaluate the performance of each machine learning algorithm, a dataset of 1,000 samples was selected and divided into training and test sets in a 7:3 ratio.

To evaluate the predictive capability of the machine learning model on unseen data, this paper employs the "train-test split" technique. First, the dataset is randomly shuffled to ensure an even distribution of data and to reduce correlation between samples. Then, the dataset is divided into a training set and a test set. The training set is used for model optimization and parameter tuning, enhancing the model's generalization ability, while the test set is reserved for the final evaluation after training is complete, to gauge its real-world performance on new data. This approach allows for a more objective measurement of the model's accuracy and robustness, providing a reliable basis for its applicability in practical scenarios. This approach insures that the model's performance is tested under unbiased conditions, providing a more accurate measure of its generalizability and effectiveness.

Table 3 shows the five metrics used to evaluate model performance: R^2, MSE, RMSE, MAE, and MRE. R^2 assesses the degree of data interpretation by the model; values close to 1 indicate good fit, while values approach 0 indicate poor fit. MSE, RMSE, MAE, and MRE measure prediction errors, with smaller values indicating higher model accuracy. Where y_i and y_i' represent the actual values and forecasted values, separately, $\bar{y}$ represents the average of the actual values in the sample, SS_{res} is the sum of squared residuals, SS_{tot} is the entire sum of squares, and N is the entire number of specimens in the dataset.

Table 3
Evaluation indexes of machine learning regression model

Name	Formulas
R^2	$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} = 1 - \frac{\sum_{i=1}^{N}(y_i - y_i')^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}$
MSE	$MSE = \frac{1}{N}\sum_{i=1}^{N}\lvert y_i - y_i'\rvert^2$
RMSE	$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\lvert y_i - y_i'\rvert^2}$
MAE	$MAE = \frac{1}{N}\sum_{i=1}^{N}\lvert y_i - y_i'\rvert$
MRE	$MRE = \frac{1}{N}\sum_{i=1}^{N}\left\lvert\frac{y_i - y_i'}{y_i}\right\rvert$

Table 4
The initial structural tension of the optimal solution (kN)

Cases	Types	CG01	CG02	CG03	XS01	XS02	XS03	HS01	HS02	HS03
1	NSGA-III-FEA	-1475	-507	-272	2059	739	545	2584	1371	323
	NSGA-III-SBO	-526	-379	-740	2779	834	404	1859	1455	387
2	NSGA-III-FEA	-1428	-683	-735	1037	807	710	3142	697	382
	NSGA-III-SBO	-1392	-510	-423	626	893	692	3147	1023	329

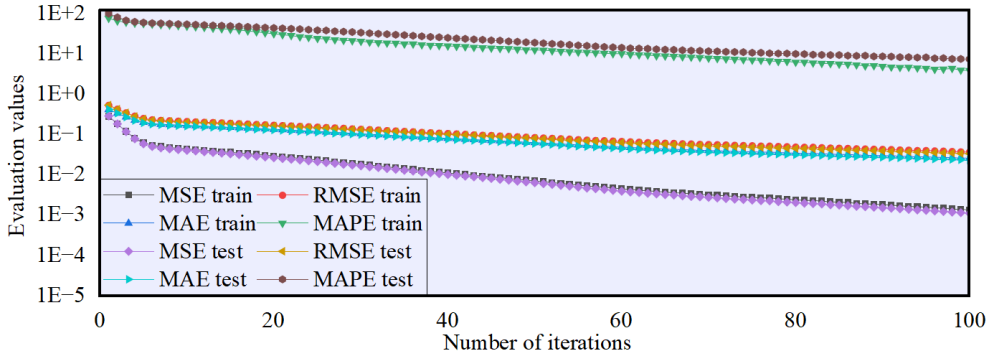
4.3. Prestress optimization

Section 3.1 introduces three neural network architectures used to build surrogate models. According to the process detailed in Section 3.2, multi-objective optimization is performed using the established surrogate models. After completing the optimization, a Pareto frontier of non-dominated solutions is obtained. Finally, the multi-criteria decision-making method described in Section 2.3 is used to select the optimal solution from the set of non-dominated solutions. Table 4 presents the prestress data for the cable-strut system obtained using both the finite element analysis software and the surrogate model.

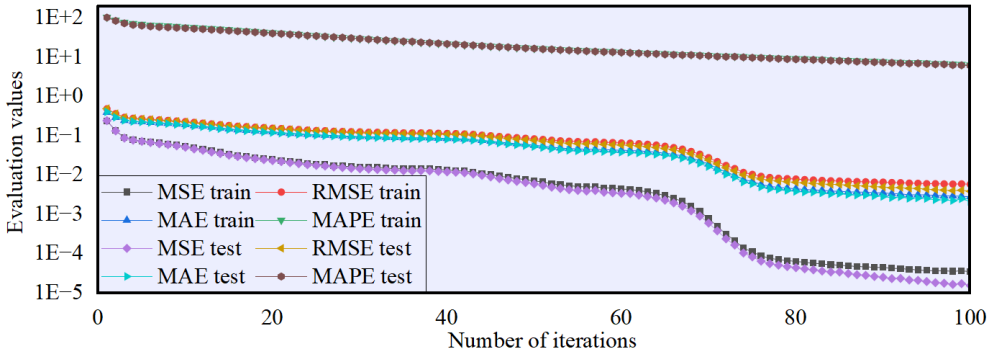
4.4. Results and discussions

Figs. 10 to 13 present the performance evaluation metrics of the three machine learning algorithms. Specifically, Figs. 10 to 12 illustrate the evolution of evaluation metrics during training for the surrogate models constructed using DBN, Seq2Seq, and BPNN, respectively. It can be observed that the MSE, MAE, MAPE, and RMSE of the dataset exhibit similar declining trends across all models, indicating substantial reductions in prediction errors as training progresses. Fig. 13 shows the model fitting performance after training. All models demonstrate high coefficients of determination (R^2), with the BPNN model achieving a value close to 1, suggesting excellent prediction accuracy.

Among the three algorithms, the BPNN-based surrogate model outperforms the others in terms of R^2, MSE, RMSE, MAE, and MAPE, indicating superior predictive capability. The R^2 values of the BPNN model on both the testing and prediction sets are close to 1, demonstrating its high reliability and effectiveness in capturing complex data relationships. This makes it particularly suitable for engineering applications with stringent accuracy requirements, such as prestress optimization in structural engineering. Therefore, the BPNN model is selected as the surrogate model for the subsequent prestress optimization study. Its optimization results are further compared with those obtained from the FEA model to validate its effectiveness.

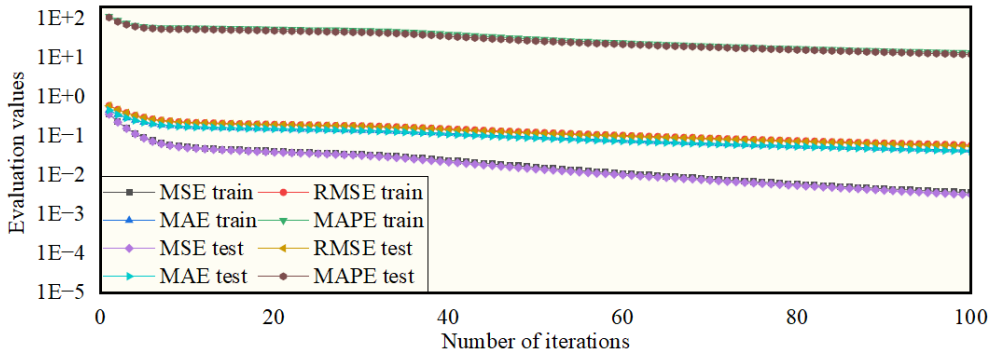


(a) The four evaluation curve of Model 1

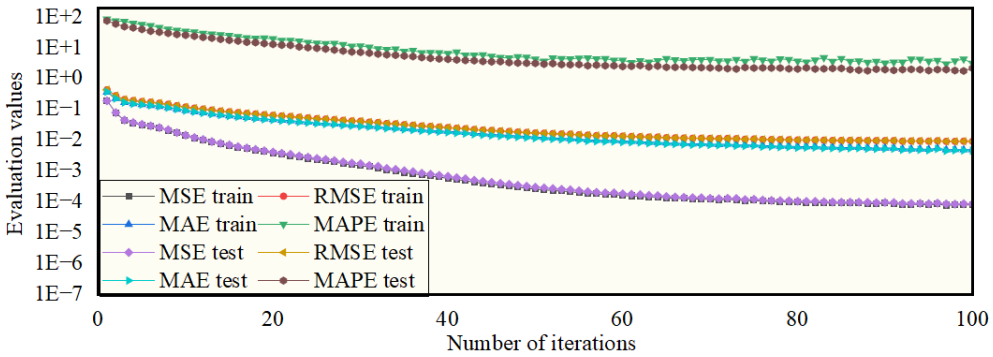


(b) The four evaluation curve of Model 2

Fig. 10 Evaluation curves of the DBN machine learning model during training and testing

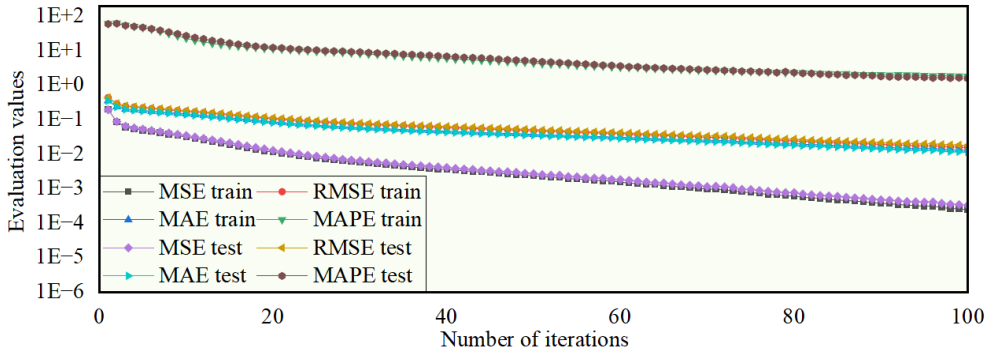


(a) The four evaluation curve of Model 1

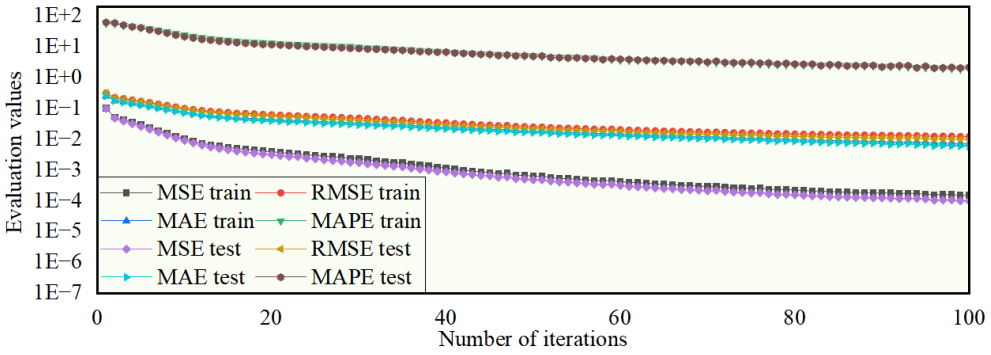


(b) The four evaluation curve of Model 2

Fig. 11 Evaluation curves of the Seq2Seq machine learning model during training and testing



(a) The four evaluation curve of Model 1



(b) The four evaluation curve of Model 2

Fig. 12 Evaluation curves of the BPNN machine learning model during training and testing

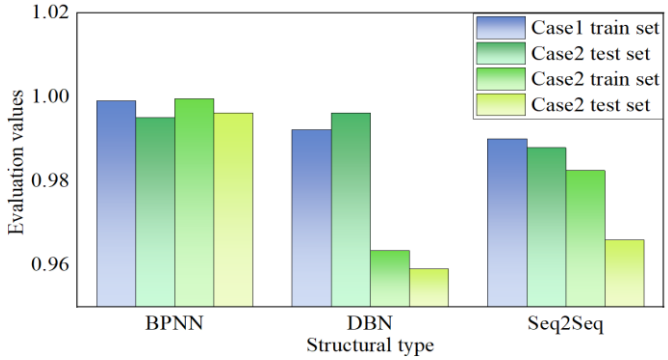
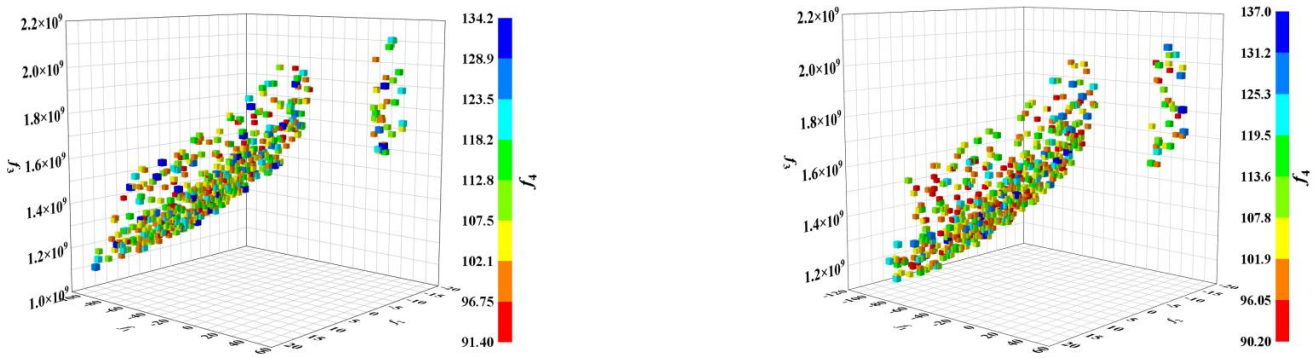


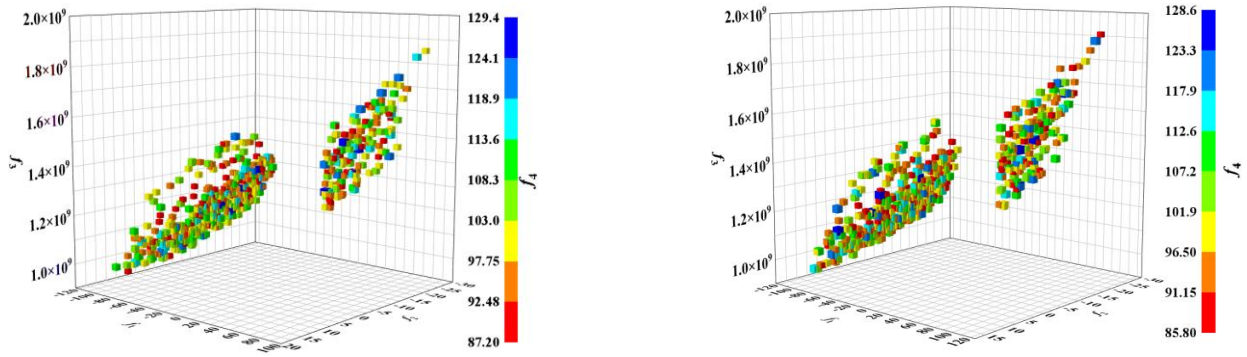
Fig. 13 The coefficient of determination (R^2) values for three different machine learning models during training and testing



(a) NSGA-III-SBO

(b) NSGA-III-FEA

Fig. 14 The pareto front of the optimized structure in Model 01



(a) NSGA-III-SBO (b) NSGA-III-FEA

Fig. 15 The pareto front of the optimized structure in Model 02

Table 5
Comparison of the best solution of various optimization algorithms

Cases	Types	$f_1(mm)$	$f_2(mm)$	f_3(J)	$f_4(t)$	Time(s)	Total time(s)
1	NSGA-III-FEA	-28.7	-6.8	1.28E9	98.5	26581	26581
	NSGA-III-SBO	-36.1	4.3	1.34E9	96.3	237	1515
2	NSGA-III-FEA	-34.7	-7.6	1.37E9	99.05	25689	25689
	NSGA-III-SBO	-31.6	-10.2	1.32E9	91.4	247	1525

Fig. 14 and Fig. 15 show multi-objective optimization results, with (a) representing optimization based on finite element analysis and (b) showing the results based on the BPNN surrogate model. In all four figures, the x, y, and z axes represent the maximum deflection of the structure (mm), the radial displacement at the supports (mm), and the minimum strain energy (J), respectively. The color scale indicates the amount of steel used in the lower cable-strut system.

Table 5 compares the optimal solutions obtained using the two methods, calculated through the AHP method for multi-criteria decision-making outlined in Section 2.3. For the surrogate model approach, the total optimization time is calculated by adding the data training time (t = 1278s) to the optimization time, resulting in $Total\ time = Time + t$. By observing Figures 10 and 11, the following conclusions can be drawn:

(1) There is a distinct linear relationship between the maximum deflection of the structure, the radial displacement at the supports, and the strain energy of the structure. The non-dominated relationships among these objectives are apparent. Strain energy and maximum vertical displacement, as well as radial displacement at the supports, show linear variation: as the support transitions from tension to compression and the maximum vertical displacement changes from negative to positive, strain energy increases.

(2) The maximum vertical displacement cannot be optimized to zero and can only reach a relatively small displacement, as the vertex of the dome structure cannot be controlled.

(3) There is no apparent relationship between the amount of steel used and the other three optimization objectives, meaning it is possible to achieve the minimum steel usage while still achieving favorable outcomes for the remaining objectives.

4.5. Limitations and future developments of the research work

This study integrates multi-objective optimization with surrogate modeling, significantly improving optimization efficiency while substantially reducing computational resource consumption. However, the proposed approach still has certain limitations, and there remains considerable room for improvement in future research:

(1) Data Dependence and Generalization: The data-driven models rely heavily on existing datasets for training and lack an understanding of underlying physical principles. As a result, their predictions can be unreliable in regions outside the training data distribution, and a large amount of data is typically required to achieve acceptable accuracy.

(2) Black-box Nature and Limited Interpretability: The adopted approach is essentially a black-box model, making it difficult to interpret the mechanisms behind the optimization results. Incorporating physical equations or domain knowledge could improve the model's transparency and interpretability.

(3) Physically Infeasible Solutions: During the optimization process, the surrogate-based model may produce solutions that violate physical constraints, such as inadequate structural strength or energy imbalance. Embedding physical boundary conditions and constraints into the model could help ensure the feasibility of the generated solutions.

(4) Lack of Sensitivity Analysis in AHP: Although the Analytic Hierarchy Process (AHP) provides a structured and logical framework for weight assignment, no sensitivity or robustness analysis was conducted in this study. Future work could address this limitation to enhance the reliability of the decision-making process.

5. Conclusion

This paper employs three different machine learning algorithms to forecast the respond of the suspen-dome structure. By inputting the initial tension, the displacement and internal force distribution of the structure were successfully predicted. Subsequently, four objective functions were constructed based on data analysis, and a multi-objective surrogate model was successfully established. On this basis, the NSGA-III algorithm was further utilized for multi-objective optimization, resulting in a set of Pareto front solutions. Finally, multi-criteria decision-making was performed using the AHP to select the optimal solution from the Pareto front.

(1) The construction of surrogate models and the options of multi-objective optimization algorithms should take into consideration both the complexity of the model and the optimization costs. Compared to other neural networks, BPNN has superior performance.

(2) The multi-objective prestress optimization method is grounded on the surrogate model not only significantly improves optimization efficiency but also more effectively coordinates the relationships among objectives, resulting in higher-quality solutions.

(3) Using the AHP to determine the best solution can effectively obtain the optimal solution from the Pareto front.

(4) This study is based on data-driven machine learning models, which inherently suffer from limited interpretability and data dependency. In addition, no sensitivity or robustness analysis was conducted on the determined weights. These limitations suggest valuable directions for future research, such as enhancing model interpretability and conducting systematic sensitivity analysis to improve the reliability of the decision-making process.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant Nos. 51308105, 52278190) and the Fundamental Research Funds for the Central Universities (Grant No. 2242022k30030).

References

[1] Kawaguchi M, Abe M, Tatemichi I. Design, tests and realization of suspen-dome system. Journal of the International Association for Shell and Spatial Structures 1999;40:179–92.
[2] Liu X. Introduction of structures of some stadiums and gymnasiums for Beijing 2008 Olympic Games. Fourth International Conference on Advances in Steel Structures, vol. I,

Elsevier; 2005, p. 361–8. https://doi.org/10.1016/B978-008044637-0/50052-4.
[3] Li Z Q, Zhang Z H, Dong S L, et al. Construction sequence simulation of a practical suspen-dome in Jinan Olympic Center. Advanced Steel Construction, 2012, 8(1): 38-53.
[4] Nie G, Fan F, Zhi X. Test on the Suspended Dome Structure and Joints of Dalian Gymnasium. Adv Struct Eng 2013;16:467–85. https://doi.org/10.1260/1369-4332.16.3.467.
[5] Gong S. The research of suspen-dome structure. IOP Conf Ser: Mater Sci Eng 2017;242:012050. https://doi.org/10.1088/1757-899X/242/1/012050.
[6] Yuan XF, Dong SL. Integral feasible prestress of cable domes. Comput Struct 2003;81:2111–9. https://doi.org/10.1016/S0045-7949(03)00254-2.
[7] Optimal Design of Cable-domes with Multi-overall Selfstress Modes-All Databases n.d. https://webofscience.clarivate.cn/wos/alldb/full-record/CSCD:3389260 (accessed September 2, 2024).
[8] Kitipornchai S, Kang W, Lam H-F, Albermani F. Factors affecting the design and construction of Lamella suspen-dome systems. Journal of Constructional Steel Research 2005;61:764–85. https://doi.org/10.1016/j.jcsr.2004.12.007.
[9] Zhang Z, Zhi X, Li Q, Fan F. Simlating the Stability and Pre-stressed Tension of a Single-Layer Spherical Reticulated Shell Supported by Cable. Int J Steel Struct 2019;19:618–34. https://doi.org/10.1007/s13296-018-0151-6.
[10] Kang W, Chen Z, Lam H-F, Zuo C. Analysis and design of the general and outmost-ring stiffened suspen-dome structures. Engineering Structures 2003;25:1685–95. https://doi.org/10.1016/S0141-0296(03)00149-4.
[11] Liang H Q, Dong S L, Miao F. Multi-objective optimization analysis of cable dome structures prestress based on niche genetric algorithm. J. Build. Struct, 2016, 37: 92-99.(in Chinese)
[12] Kaveh A, Rezaei M. Optimal Design of Double-Layer Domes Considering Different Mechanical Systems via ECBO. Iran J Sci Technol Trans Civ Eng 2018;42:333–44. https://doi.org/10.1007/s40996-018-0123-2.
[13] Olofin I, Liu R. Suspen-Dome System: A Fascinating Space Structure. TOCIEJ 2017;11:131–42. https://doi.org/10.2174/1874149501711010131.
[14] Lygoe RJ, Cary M, Fleming PJ. A Real-World Application of a Many-Objective Optimisation Complexity Reduction Process. In: Purshouse RC, Fleming PJ, Fonseca CM, Greco S, Shaw J, editors. Evolutionary Multi-Criterion Optimization, vol. 7811, Berlin, Heidelberg: Springer Berlin Heidelberg; 2013, p. 641–55. https://doi.org/10.1007/978-3-642-37140-0_48.
[15] Deb K, Jain H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. IEEE Transactions on Evolutionary Computation 2013;18:577–601. https://doi.org/10.1109/TEVC.2013.2281535.
[16] Hisao Ishibuchi, Noritaka Tsukamoto, Yusuke Nojima. Evolutionary many-objective optimization: A short review. 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), Hong Kong, China: IEEE; 2008, p. 2419–26. https://doi.org/10.1109/CEC.2008.4631121.
[17] Ma Q, Ohsaki M, Chen Z, Yan X. Multi-objective optimization for prestress design of cable-strut structures. International Journal of Solids and Structures 2019;165:137–47. https://doi.org/10.1016/j.ijsolstr.2019.01.035.
[18] Chen Y, Yan J, Sareh P, Feng J. Feasible Prestress Modes for Cable-Strut Structures with Multiple Self-Stress States Using Particle Swarm Optimization. J Comput Civ Eng 2020;34:04020003. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000882.
[19] Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Computat 2002;6:182–97. https://doi.org/10.1109/4235.996017.
[20] Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the strength Pareto evolutionary algorithm. TIK Report 2001;103.
[21] Purshouse RC, Fleming PJ. Evolutionary many-objective optimisation: An exploratory analysis. The 2003 Congress on Evolutionary Computation, 2003. CEC'03., vol. 3, IEEE; 2003, p. 2066–73. https://doi.org/10.1109/CEC.2003.1299927.
[22] Hughes EJ. Evolutionary many-objective optimisation: many once or one many? 2005 IEEE congress on evolutionary computation, vol. 1, IEEE; 2005, p. 222–7. https://doi.org/10.1109/CEC.2005.1554688.
[23] Deb K, Mohan M, Mishra S. Evaluating the ϵ-domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions. Evolutionary Computation 2005;13:501–25. https://doi.org/10.1162/106365605774666895.
[24] Yang S, Li M, Liu X, Zheng J. A grid-based evolutionary algorithm for many-objective optimization. IEEE Transactions on Evolutionary Computation 2013;17:721–36. https://doi.org/10.1109/TEVC.2012.2227145.
[25] He Z, Yen GG, Zhang J. Fuzzy-based Pareto optimality for many-objective evolutionary algorithms. IEEE Transactions on Evolutionary Computation 2013;18:269–85. https://doi.org/10.1109/TEVC.2013.2258025.
[26] Carleo G, Cirac I, Cranmer K, Daudet L, Schuld M, Tishby N, et al. Machine learning and the physical sciences. Rev Mod Phys 2019;91:045002. https://doi.org/10.1103/RevModPhys.91.045002.
[27] Chang C-M, Lin T-K, Chang C-W. Applications of neural network models for structural health monitoring based on derived modal properties. Measurement 2018;129:457–70. https://doi.org/10.1016/j.measurement.2018.07.051.
[28] Amini K, Jalalpour M, Delatte N. Advancing concrete strength prediction using non-destructive testing: Development and verification of a generalizable model. Construction and Building Materials 2016;102:762–8. https://doi.org/10.1016/j.conbuildmat.2015.10.131.
[29] Zoph B. Neural architecture search with reinforcement learning. arXiv preprint arXiv: 1611.01578, 2016.
[30] Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative Adversarial Networks. Communications of the ACM, 2020, 63(11): 139-144.
[31] Tai L, Zhang J, Liu M, Boedecker J, Burgard W. A Survey of Deep Network Solutions for Learning Control in Robotics: From Reinforcement to Imitation. arXiv preprint arXiv:1612.07139, 2016.
[32] Li Y, Hariri-Ardebili MA, Deng T, Wei Q, Cao M. A surrogate-assisted stochastic optimization inversion algorithm: Parameter identification of dams. Adv Eng Inform 2023;55:101853. https://doi.org/10.1016/j.aei.2022.101853.
[33] Zhou Y, Zheng S, Zhang G. Machine learning-based optimal design of a phase change material integrated renewable system with on-site PV, radiative cooling and hybrid ventilations—study of modelling and application in five climatic regions. Energy 2020;192:116608. https://doi.org/10.1016/j.energy.2019.116608.
[34] JGJ 7-2010. Technical Specification for Space Grid Structures, 2010.(in Chinese)
[35] Hinton GE, Osindero S, Teh Y-W. A Fast Learning Algorithm for Deep Belief Nets. Neural Computation 2006;18:1527–54. https://doi.org/10.1162/neco.2006.18.7.1527.
[36] Vinyals O, Le Q. A neural conversational model. arXiv preprint arXiv:1506.05869, 2015.
[37] Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. Nature 1986;323:533–6. https://doi.org/10.1038/323533a0.
[38] Gu Q, Wang R, Xie H, Li X, Jiang S, Xiong N. Modified non-dominated sorting genetic algorithm III with fine final level selection. Appl Intell 2021;51:4236–69. https://doi.org/10.1007/s10489-020-02053-z.
[39] Deb K, Jain H. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. IEEE Trans Evol Computat 2014;18:577–601. https://doi.org/10.1109/TEVC.2013.2281535.