Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2012, Article ID 712974, 18 pages
doi:10.1155/2012/712974

Research Article

Prediction of Optimal Design and Deflection of Space Structures Using Neural Networks

Reza Kamyab Moghadas,[1, 2] Kok Keong Choong,[2] and Sabarudin Bin Mohd[3]

[1] *The Iranian Academic Center for Education, Culture and Research, Kerman 7616914111, Iran*
[2] *School of Civil Engineering, Universiti Sains Malaysia, Nibong Tebal, Penang, Malaysia*
[3] *City University College, Petaling Jaya, Selangor, Malaysia*

Correspondence should be addressed to Reza Kamyab Moghadas, r_kamyab_m@yahoo.com

Received 8 October 2012; Accepted 18 November 2012

Academic Editor: Siamak Talatahari

The main aim of the present work is to determine the optimal design and maximum deflection of double layer grids spending low computational cost using neural networks. The design variables of the optimization problem are cross-sectional area of the elements as well as the length of the span and height of the structures. In this paper, a number of double layer grids with various random values of length and height are selected and optimized by simultaneous perturbation stochastic approximation algorithm. Then, radial basis function (RBF) and generalized regression (GR) neural networks are trained to predict the optimal design and maximum deflection of the structures. The numerical results demonstrate the efficiency of the proposed methodology.

1. Introduction

The history of the applications of Artificial Intelligence to civil and structural engineering is simultaneously brief and long. It is brief if compared to the history of civil and structural engineering, whose definition as a discipline can be fixed a very long time ago. It makes sense to consider civil and structural engineering as the most ancient applicative discipline, being founded in preclassical world by Egyptians and Babylonians. It is long, instead, if compared to the history of Artificial Intelligence, whose name first appeared in science at the end of the sixties of the twentieth century. The earliest applications to civil and structural engineering are very likely [1], where authors review tools and techniques for knowledge-based expert system for engineering design. An even earlier paper whose scope was indeed wider, but introduced some fundamental themes, is [2]. We can definitely settle a start date in 1986 when the first International Symposium on this theme took place [3]. In statistical terms, since we can fix Artificial Intelligence engineering applications start date in 1950 when the first attempt to provide a true intelligent program was carried out [4], this means that, nowadays, we are experiencing 18 years of history of the applications of Artificial Intelligence techniques to civil and structural engineering.

As in this study our main aim is to employ neural networks to predict the optimal design and maximum deflection of the double layer grids, the next paragraph is devoted to review the literature about the optimal design of space structures by soft computing techniques.

Erbatur et al. [5] reported the development of a computer-based systematic approach for discrete optimal design of planar and space structures composed of one-dimensional elements. Rajasekaran [6] created the input for large space structures using Formian. In this paper, a new optimization technique called cellular automata (CA) has been combined with genetic algorithm (GA) to develop different search and optimization known as cellular genetic algorithm (CGA), which considers the areas of the space structures as discrete variables. Krishnamoorthy et al. [7] proposed GA with objective-oriented framework which was used in space truss optimization. Tashakori and Adeli [8] adopted the patented robust neural dynamics model for optimum design of space trusses made of commercially available cold-formed shapes in accordance with AISC specification. E. Salajegheh and J. Salajegheh [9] achieved the optimal design of space structures while the design variables are continuous and discrete. To reduce the computational work involved in the optimization process they employed a semiquadratic function; also they use a hybrid form of the approximation. Kaveh and Dehkordi [10] trained neural networks for the analysis, design, and prediction of the displacements of domes using the backpropagation and radial basis Functions neural networks. The performance of these networks is compared when applied to domes. Kaveh et al. [11] combined the energy method and the force method in the context of transmission tower optimization in order to form a holistic design and optimization approach, eliminating the need for time-intensive matrix inversion. The addition of a neural network as an analysis tool reduces the overall computational load. Kaveh and Servati [12] trained neural networks for design of square diagonal-on-diagonal double layer grids. They employed backpropagation algorithm for training the networks for evaluation of the maximum deflection, weight, and design of the double layer grids. Salajegheh and Gholizadeh [13] employed a modified genetic algorithm (GA) and radial basis function (RBF) neural networks to optimize space structures. Kaveh et al. [14] employed ant colony optimization (ACO) algorithm for optimal design of space structures with fixed geometry. Gholizadeh et al. [15] employed a combination of GA and wavelet radial basis function (WRBF) neural networks to find the optimal weight of structures subject to multiple natural frequency constraints.

Much more other applications of neural networks in the field of civil engineering can be found in the literature [16–20].

In this investigation, an innovative methodology is proposed to predict the optimal design and maximum deflection of the square-on-square double layer grids. This methodology consists of three stages. In the first stage, a number of the double layer grids with random spans and heights are generated. In the second stage the generated double layer grids are optimized by an optimization algorithm. Although, in the recent years many new structural optimization algorithms have been proposed by the researchers [21–25], in this paper, simultaneous perturbation stochastic approximation (SPSA) [26] algorithm is used due to its computational merits. Also, the maximum deflections of the optimal structures are saved. In the third stage, radial basis function (RBF) [27] and generalized regression (GR) [27] neural networks are trained to predict the optimal design and maximum deflection of the double layer grids. To design neural networks MATLAB [28] is employed.

2. Formulation of Optimization Problem

In optimal design problem of space trusses the aim is to minimize the weight of the truss under constraints on stresses and displacements. This optimization problem can be expressed as follows:

$$\text{minimize:} \quad w(x_1, \dots, x_n, \dots, x_{ng}) = \sum_{n=1}^{ng} x_n \sum_{m=1}^{nm} \gamma_m l_m, \tag{2.1}$$

$$\begin{aligned} \text{subject to:} \quad & \sigma_i \le \sigma_{\text{all},i}, \quad i = 1, 2, \dots, ne, \\ & \delta_j \le \delta_{\text{all},j,} \quad j = 1, 2, \dots, nj, \end{aligned} \tag{2.2}$$

where x_n, γ_m, and l_m are cross-sectional area of members belonging to group n, weight density, and length of mth element in this group, respectively; ng and nm are the total number of groups in the structure and the number of members in group n, respectively; ne and nj are the total number of the elements and nodes in truss, respectively; σ_i and δ_j are stress in the ith element and displacement of the jth node, respectively. Also, $\sigma_{\text{all},i}$ and $\delta_{\text{all},j}$ are allowable stress in the ith member and allowable deflection of the jth node, respectively.

In this study, besides cross-sectional areas (x_n) the geometry dependent parameters of the double layer grid, L and h, are also variables. In other words, the aim is to find optimal cross-sectional areas for each set of L and h. Thus, (2.1) can be reexpressed as follows:

$$\text{For each set of } L \text{ and } h \text{ minimize } w(x_1, \dots, x_n, \dots, x_{ng}) = \sum_{n=1}^{ng} x_n \sum_{m=1}^{nm} \gamma_m l_m. \tag{2.3}$$

It is obvious that the computational burden of the above optimization problem is very high due to the fact that L and h are variables. Employing the neural network technique can substantially reduce the computational costs.

As the SPSA requires less number of function evaluations (structural analyses) than the other type of gradient-based methods, it is selected as the optimizer in this study. The basic concepts of the SPSA are explained in the next section.

3. SPSA Optimization Algorithm

SPSA has recently attracted considerable international attention in areas such as statistical parameter estimation, feedback control, simulation-based optimization, signal and image processing, and experimental design. The essential feature of SPSA is the underlying gradient approximation that requires only two measurements of the objective function regardless of the dimension of the optimization problem. This feature allows for a significant reduction in computational effort of optimization, especially in problems with a large number of variables to be optimized. The basic unconstrained SPSA optimization algorithm is in the general recursive stochastic approximation (SA) form [26]:

$$\hat{X}_{k+1} = \hat{X}_k - a_k \hat{G}_k\left(\hat{X}_k\right), \tag{3.1}$$

where $\hat{X}_k$ represents the estimate of X at kth iteration, $a_k > 0$ represent a scalar gain coefficient, and $\hat{G}(\hat{X}_k)$ represent an approximate gradient at $\hat{X}_k$. Under appropriate condition, (3.1) will converge to optimum design X^* in some stochastic sense. The essential part of (3.1) is the gradient approximation that is obtained using the simultaneous perturbation (SP) method. Let $w(\cdot)$ denote a measurement of objective function at a design level represented by the dot and let c_k be some positive number. The SP approximation has all elements of $\hat{X}_k$ randomly perturbed together to obtain two measurements of $w(\cdot)$, but each component is formed from a ratio involving the individual components in the perturbation vector and the difference in the two corresponding measurement. For two sided simultaneous perturbation, we have

$$G_{ki}\left(\hat{X}_k\right) = \frac{w\left(\hat{X}_k + c_k \Delta_{ki}\right) - w\left(\hat{X}_k - c_k \Delta_{ki}\right)}{2 c_k \Delta_{ki}}, \tag{3.2}$$

where the distribution of the user-specified n_v dimensional random perturbation vector $\Delta_k = \{\Delta_{k1}, \Delta_{k2}, \ldots, \Delta_{kn_v}\}^T$ satisfies condition discussed in [26].

It is observed that each iteration of SPSA needs only two objective function measurements independent of n_v because the numerator is the same in all n_v components. This circumstance provides the potential for SPSA to achieve a large savings in the total number of measurements required to estimate X^* when n_v is large.

3.1. Implementation of SPSA

The following step-by-step summary shows how SPSA iteratively produces a sequence of estimates [26].

Step 1 (initialization and coefficient selection). Set counter index $k = 0$. Pick initial guess and nonnegative coefficients a, c, A, α, and γ in the SPSA gain sequences $a_k = a/(A + k + 1)^{\alpha}$ and $c_k = c/(k+1)^{\gamma}$. The choice of gain sequences (a_k and c_k) is critical to the performance of SPSA. Spall provides some guidance on picking these coefficients in a practically manner.

Step 2 (generation of the simultaneous perturbation vector). Generate by Monte Carlo an n_v-dimensional random perturbation vector Δ_k, where each of the n_v components of Δ_k is independently generated from a zero mean probability distribution satisfying some conditions. A simple choice for each component of Δ_k is to use a Bernoulli ±1 distribution with probability of 1/2 for each ±1 outcome. Note that uniform and normal random variables are not allowed for the elements of Δ_k by the SPSA regularity conditions.

Step 3 (objective function evaluations). Obtain two measurements of the objective function $w(\cdot)$ based on simultaneous perturbation around the current $\hat{X}_k : w(\hat{X}_k + c_k\Delta_k)$ and $w(\hat{X}_k - c_k\Delta_k)$ with the c_k and Δ_k from Steps 1 and 2.

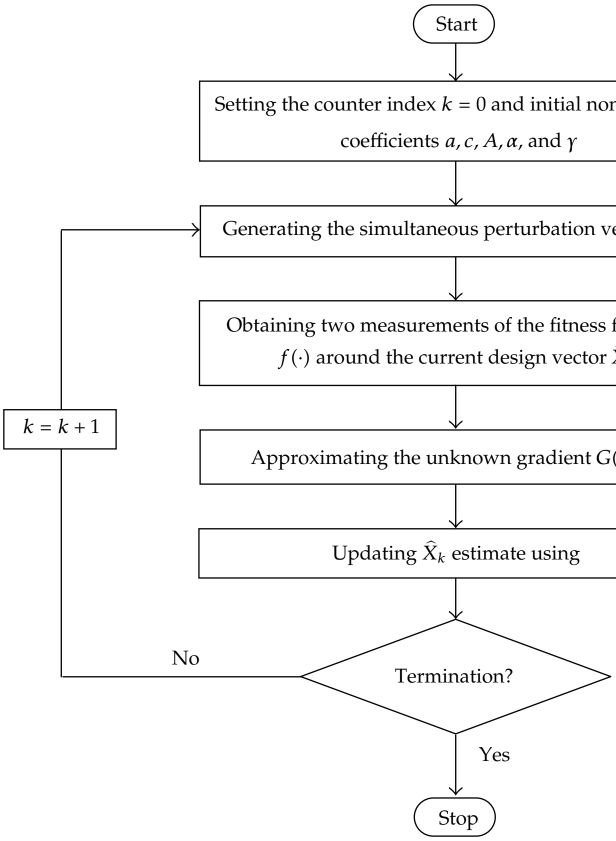

Start
Setting the counter index $k = 0$ and initial nonnegative coefficients a, c, A, α, and γ
Generating the simultaneous perturbation vector Δ_k
Obtaining two measurements of the fitness function $f(\cdot)$ around the current design vector $\widehat{X}_k$
$k = k + 1$
Approximating the unknown gradient $G(\widehat{x}_k)$
Updating $\widehat{X}_k$ estimate using
No
Termination?
Yes
Stop

Figure 1: Flowchart of the SPSA.

Step 4 (gradient approximation). Generate the simultaneous perturbation approximation to the unknown gradient $G(\widehat{X}_k)$:

$$\widehat{G}_k\left(\widehat{X}_k\right) = \frac{w\left(\widehat{X}_k + c_k\Delta_k\right) - w\left(\widehat{X}_k - c_k\Delta_k\right)}{2c_k}\begin{bmatrix}\Delta_{k1}^{-1}\\ \Delta_{k2}^{-1}\\ \vdots\\ \Delta_{kn_v}^{-1}\end{bmatrix}, \tag{3.3}$$

where Δ_{ki} is the ith component of Δ_k vector.

Step 5 (updating X estimate). Use the standard SA to update $\widehat{X}_k$ to new value $\widehat{X}_{k+1}$.

Step 6 (iteration or termination). Return to Step 2 with $k + 1$ replacing k. Terminate the algorithm if there is little change in several successive iterates or the maximum allowable number of iterations has been reached. Figure 1 shows the flowchart of the SPSA.

In the present work, we suppose that the length and height of the double layer grids are varied in specific ranges. Our aim is to optimize all of the possible structures defined in the ranges. Therefore it can be observed that the additional difficulty is the huge computational burden of the optimization process. In order to mitigate the difficulty, RBF and GR neural networks are employed to predict the optimal design of the double layer grids with various length and height.

4. Neural Networks

In the recent years, neural networks are considered as more appropriate techniques for simplification of complex and time consuming problems. The interest shown to neural networks is mainly due to their ability to process and map external data and information based on past experiences. Neural networks are not programmed to solve specific problems. Indeed, neural networks never use rules or physic equations related to the specific problem in which they are employed. Neural networks use the knowledge gained from past experiences to adapt themselves to solve the new problems.

4.1. Radial Basis Function

The use of RBF in the design of neural networks was first introduced by Wasserman in 1993 [27]. The RBF network basically involves three entirely different layers: an input layer, a hidden layer of high enough dimension, and an output layer. The transformation from the hidden unit to the output space is *linear*. Each output node is the weighted sums of the outputs of the hidden layer. However, the transformation from the input layer to the hidden layer is *nonlinear*. Each neuron or node in the hidden layer forming a linear combination of the basis (or kernel) functions which produces a localized response with respect to the input signals. This is to say that RBF produce a significant nonzero response only when the input falls within a small localized region of the input space. The most common basis of the RBF is a Gaussian kernel function of the form:

$$\varphi_l(Z) = \exp\left[-\frac{(Z - C_l)^T (Z - C_l)}{2\sigma_l}\right], \quad l = 1, 2, \ldots, q, \tag{4.1}$$

where φ_1 is the output of the lth node in hidden layer; Z is the input pattern; C_l is the weight vector for the lth node in hidden layer, that is, the center of the Gaussian for node l; σ_l is the normalization parameter (the measure of spread) for the lth node; and q is the number of nodes in the hidden layer. The outputs are in the range from zero to one so that the closer the input is to the center of the Gaussian, the larger the response of the node is. The name RBF comes from the fact that these Gaussian kernels are radially symmetric; that is, each node produces an identical output for inputs that lie a fixed radial distance from the center of the kernel C_l. The network outputs are given by

$$y_i = Q_i^T \varphi_l(Z), \quad i = 1, 2, \ldots, M, \tag{4.2}$$

where y_i is the output of the ith node, Q_i is the weight vector for this node, and M is the number of nodes in the output layer.

There are two common ways to calculate the measure of spread σ_l.

(1) Find the measure of spread from the set of all training patterns grouped with each cluster center C_l; that is, set them equal to the average distance between the cluster centers and the training patterns:

$$\sigma_l^2 = \frac{1}{N_l} \sum_{k \in C_l} (Z_k - C_l)^T (Z_k - C_l), \qquad l = 1, 2, \ldots, q, \tag{4.3}$$

where N_l is the number of patterns that belong to the lth cluster and k is the index number of a pattern that belongs to the lth cluster.

(2) Find the measure of spread from among the centers (p-nearest neighbor heuristic):

$$\sigma_l^2 = \frac{1}{N_l} \sum_{k=1}^{q} (C_k - C_l)^T (C_k - C_l), \quad l = 1, 2, \ldots, q. \tag{4.4}$$

4.2. Generalized Regression

Generalized regression network (GR) subsumes the basis function methods. This network does not require iterative training. The structure of GR is designated such that transpose of input matrix and transpose of desired output (target) matrix are chosen as first layer and second layer weight matrices, respectively. GR algorithm is based on nonlinear regression theory, a well established statistical technique for function estimation. Except the approach of adjusting of second layer weights, the other aspects of GR are identical to RBF neural networks.

5. Proposed Methodology

5.1. Double Layer Grid Model

In this section dimensions of considered double layer grid structure and its corresponding model are described. The model considered here is a double layer grid with bar elements connected by pin joints. The length of the spans, L, is varied between 25 and 75 m with step of 5 m. The height is varied between 0.035 and 0.095 L with steps of 0.2 m. The smallest and biggest structures in this interval are shown in Figure 2. The sum of dead and live loads equal to 250 kg/m^2 is applied to the nodes of the top layer.

In order to satisfy practical demands, in the optimization of large-scaled structure such as space structures, the structural elements should be divided into some groups. In this study the elements are put into 18 different groups. For this purpose a step-by-step summary defined bellow is employed.

Step 1. A similar cross sectional area is initially assigned to all elements of the structure.

Step 2. The structure is analyzed through FE and axial stresses of all members are obtained.

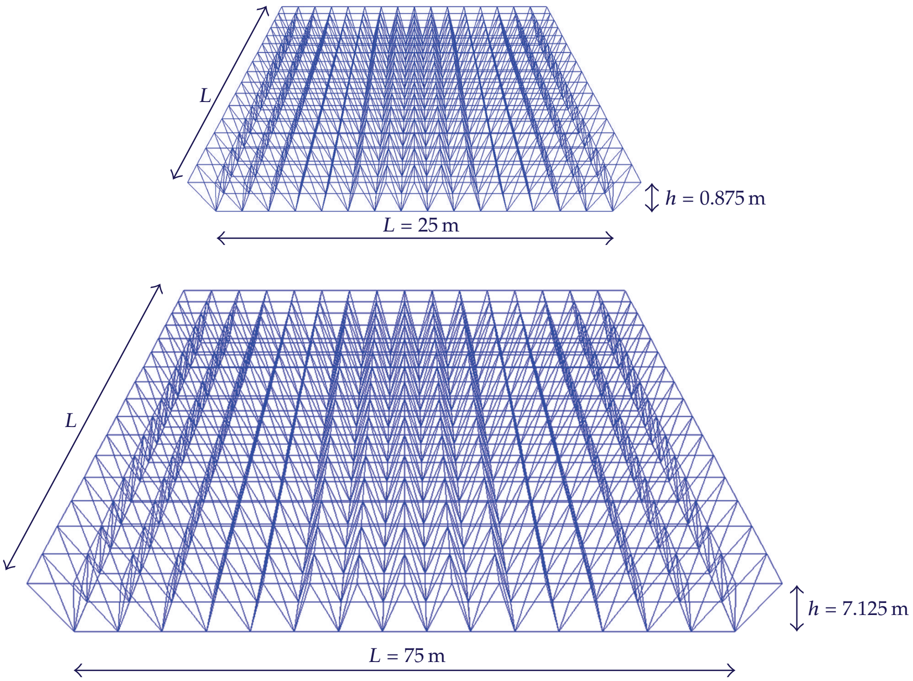


Figure 2: The smallest and biggest structures in the considered interval.

Step 3. All tension members of the structure are put into 6 groups according to their stress states as follows:

$$\begin{aligned}
0.00 \leq \sigma < 200\,\mathrm{kg/cm^2} &\Longrightarrow \text{group } 1,\\
200 \leq \sigma < 400\,\mathrm{kg/cm^2} &\Longrightarrow \text{group } 2,\\
400 \leq \sigma < 600\,\mathrm{kg/cm^2} &\Longrightarrow \text{group } 3,\\
600 \leq \sigma < 800\,\mathrm{kg/cm^2} &\Longrightarrow \text{group } 4,\\
800 \leq \sigma < 1000\,\mathrm{kg/cm^2} &\Longrightarrow \text{group } 5,\\
\sigma \geq 1000\,\mathrm{kg/cm^2} &\Longrightarrow \text{group } 6.
\end{aligned} \tag{5.1}$$

Step 4. All compressive members of top and bottom layer elements of structure are put into 6 deferent groups according to their stress values as follows:

$$\begin{aligned}
-200 \leq \sigma < -0.00\,\mathrm{kg/cm^2} &\Longrightarrow \text{group } 7,\\
-400 \leq \sigma < -200\,\mathrm{kg/cm^2} &\Longrightarrow \text{group } 8,\\
-600 \leq \sigma < -400\,\mathrm{kg/cm^2} &\Longrightarrow \text{group } 9,\\
-800 \leq \sigma < -600\,\mathrm{kg/cm^2} &\Longrightarrow \text{group } 10,
\end{aligned}$$

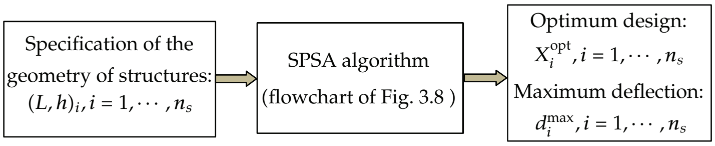

Specification of the geometry of structures: $(L,h)_i, i = 1,\cdots,n_s$
SPSA algorithm (flowchart of Fig. 3.8)
Optimum design: $X_i^{\text{opt}}, i = 1,\cdots,n_s$
Maximum deflection: $d_i^{\max}, i = 1,\cdots,n_s$

Figure 3: Data generation process.

$$
\begin{gathered}
-1000 \leq \sigma < -800\,\text{kg/cm}^2 \Longrightarrow \text{group } 11,\\
\sigma < -1000\,\text{kg/cm}^2 \Longrightarrow \text{group } 12.
\end{gathered}
\tag{5.2}
$$

Step 5. All compressive members of middle layer elements of structure are also put into 6 deferent groups based on their stresses as follows:

$$
\begin{gathered}
-200 \leq \sigma < -0.00\,\text{kg/cm}^2 \Longrightarrow \text{group } 13,\\
-400 \leq \sigma < -200\,\text{kg/cm}^2 \Longrightarrow \text{group } 14,\\
-600 \leq \sigma < -400\,\text{kg/cm}^2 \Longrightarrow \text{group } 15,\\
-800 \leq \sigma < -600\,\text{kg/cm}^2 \Longrightarrow \text{group } 16,\\
-1000 \leq \sigma < -800\,\text{kg/cm}^2 \Longrightarrow \text{group } 17,\\
\sigma < -1000\,\text{kg/cm}^2 \Longrightarrow \text{group } 18.
\end{gathered}
\tag{5.3}
$$

Preparing a neural network is achieved in three stages: data generating, training, and testing. In the first stage, a number of input and output pairs are provided and divided into training and testing sets. In the second stage, the training set is used and the modifiable parameters of the neural network are adjusted. In the last stage the performance generality of the trained neural network is examined through the testing set.

In order to provide the required data (data generation), a number of double layer grids according to their L and h are randomly selected. All of the selected structures are optimized using SPSA. Optimal designs of the selected structures and their corresponding maximum deflections are saved. This process is shown in Figure 3.

In order to train neural networks, the generated data should be separated to n_{tr} training data and n_{ts} testing data ($n_{\text{tr}} + n_{\text{ts}} = n_s$) as follows.

Training data for optimal design predictor networks:

$$
\text{Training data}: \begin{cases} \text{Inputs: } (L,h)_i, \\ \text{Outputs: } X_i^{\text{opt}} = \begin{bmatrix} x_1^{\text{opt}} \\ x_2^{\text{opt}} \\ \vdots \\ x_{18}^{\text{opt}} \end{bmatrix}_i, \quad i = 1,2,\ldots,n_{\text{tr}}. \end{cases}
\tag{5.4}
$$

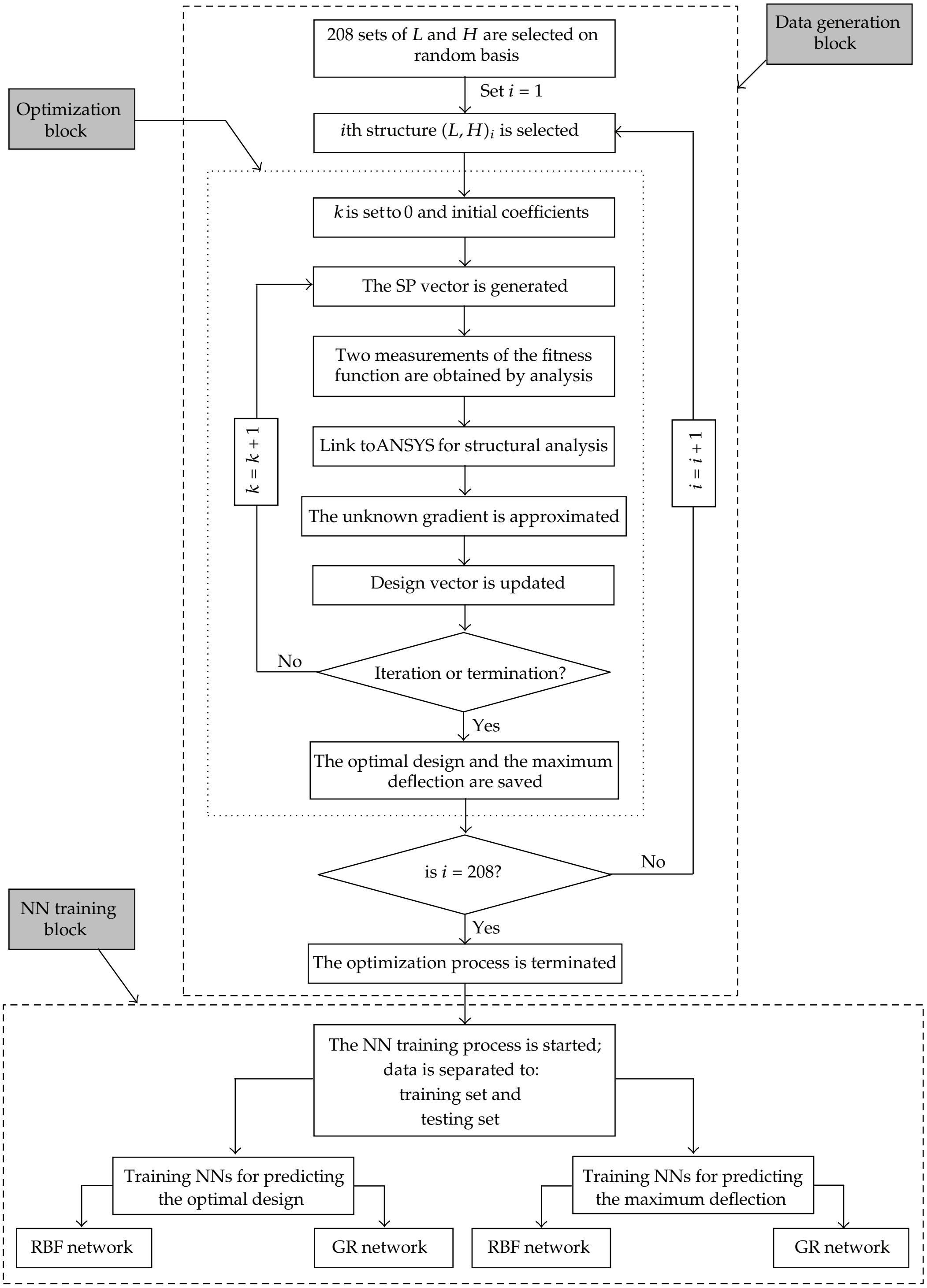

Data generation block
Optimization block
NN training block
208 sets of L and H are selected on random basis
Set $i = 1$
ith structure $(L, H)_i$ is selected
k is set to 0 and initial coefficients
The SP vector is generated
Two measurements of the fitness function are obtained by analysis
Link to ANSYS for structural analysis
The unknown gradient is approximated
Design vector is updated
Iteration or termination?
No
$k = k + 1$
Yes
The optimal design and the maximum deflection are saved
is $i = 208$?
No
$i = i + 1$
Yes
The optimization process is terminated
The NN training process is started; data is separated to: training set and testing set
Training NNs for predicting the optimal design
Training NNs for predicting the maximum deflection
RBF network
GR network
RBF network
GR network

Figure 4: Flowchart of the proposed methodology.

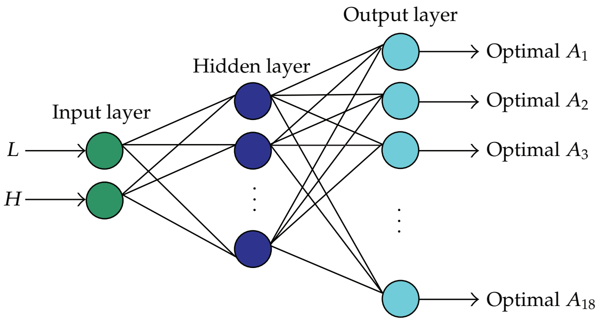


Figure 5: Typical topology of a neural network model to predict the optimal design.

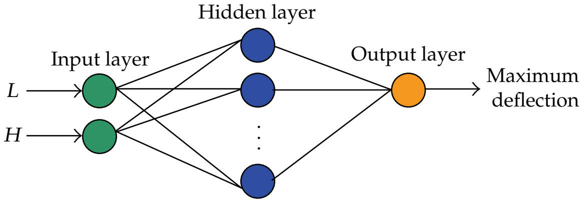


Figure 6: Typical topology of a neural network model to predict the maximum deflection.

Training data for maximum deflection predictor networks:

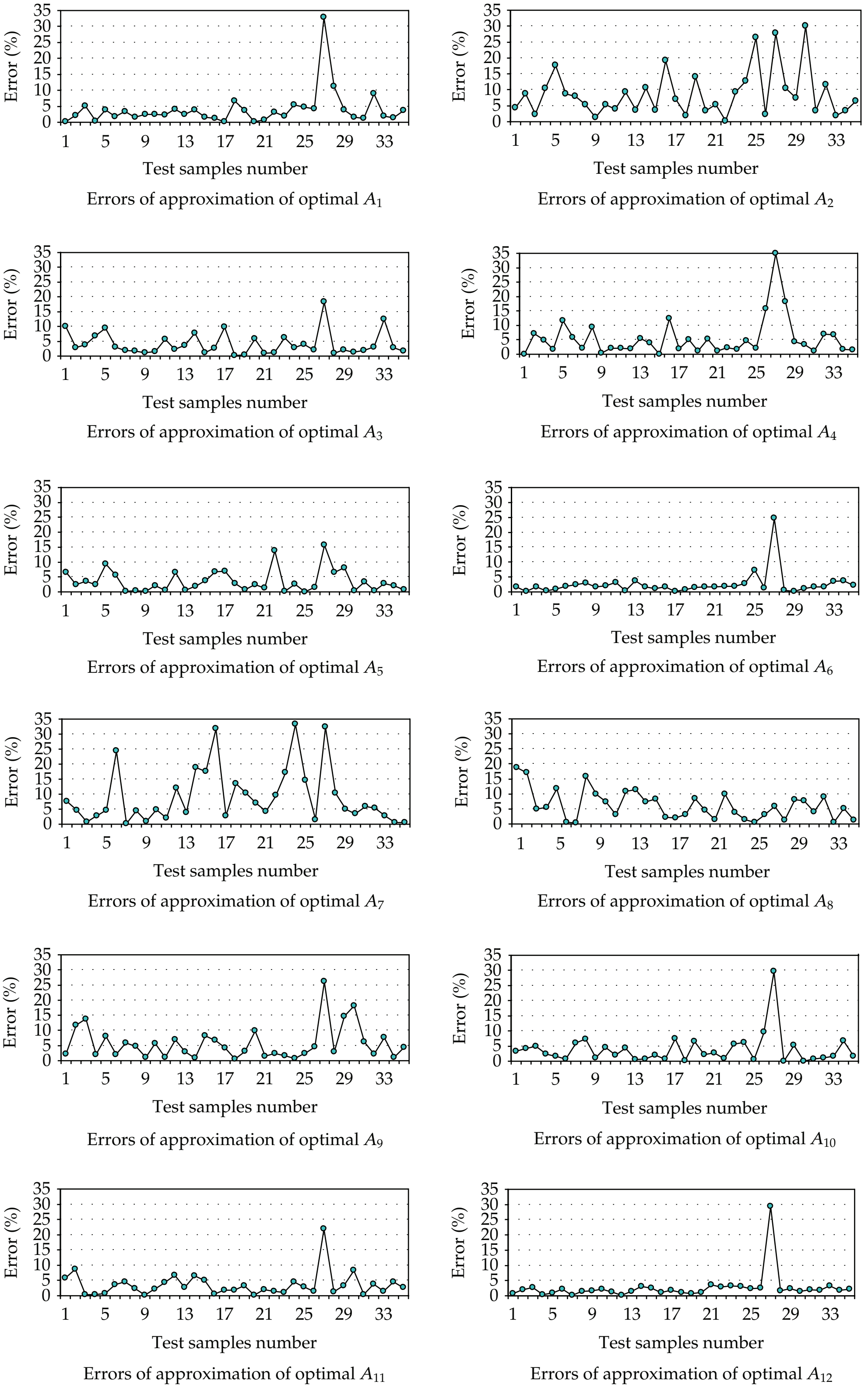
$$\text{Training data}: \begin{cases} \text{Inputs: } (L, h)_i, \\ \text{Outputs: } d_i^{\max}, \quad i = 1, 2, \ldots, n_{\text{tr}}. \end{cases} \tag{5.5}$$

Testing data for optimal design predictor networks:

$$\text{Testing data}: \begin{cases} \text{Inputs: } (L, h)_i, \\ \text{Outputs: } \mathrm{X}_i^{\text{opt}} = \begin{bmatrix} x_1^{\text{opt}} \\ x_2^{\text{opt}} \\ \vdots \\ x_{18}^{\text{opt}} \end{bmatrix}_i, \quad i = 1, 2, \ldots, n_{\text{ts}}. \end{cases} \tag{5.6}$$

Testing data for maximum deflection predictor networks:

$$\text{Testing data}: \begin{cases} \text{Inputs: } (L, h)_i, \\ \text{Outputs: } d_i^{\max}, \quad i = 1, 2, \ldots, n_{\text{ts}}. \end{cases} \tag{5.7}$$



Errors of approximation of optimal A_1

Errors of approximation of optimal A_2

Errors of approximation of optimal A_3

Errors of approximation of optimal A_4

Errors of approximation of optimal A_5

Errors of approximation of optimal A_6

Errors of approximation of optimal A_7

Errors of approximation of optimal A_8

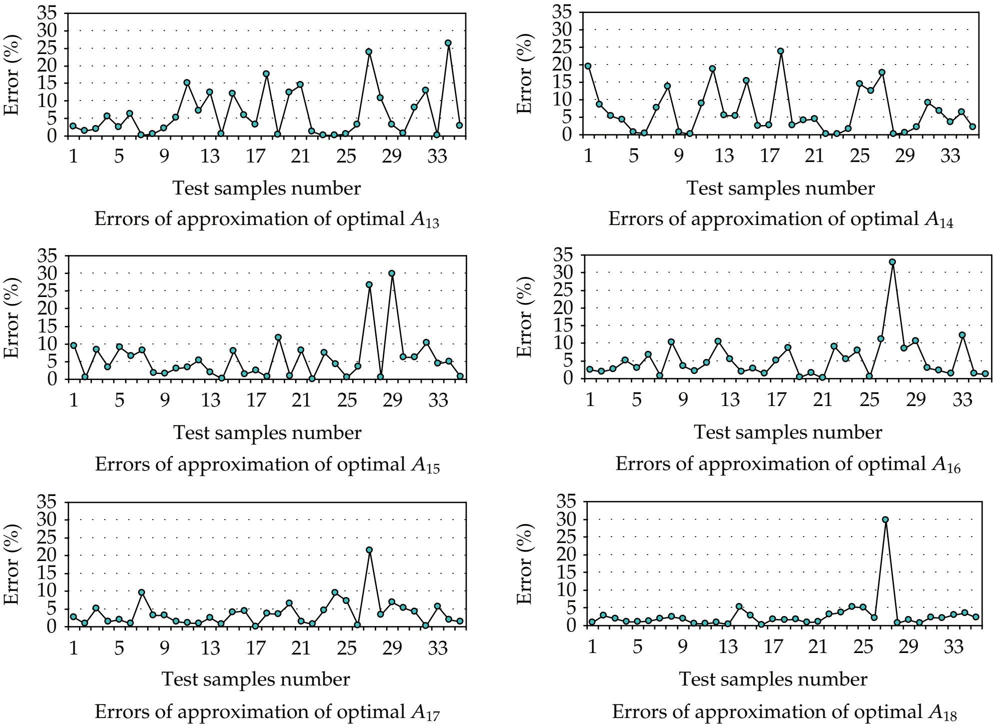
Errors of approximation of optimal A_9

Errors of approximation of optimal A_{10}

Errors of approximation of optimal A_{11}

Errors of approximation of optimal A_{12}

Figure 7: Continued.



Errors of approximation of optimal A_{13}

Errors of approximation of optimal A_{14}

Errors of approximation of optimal A_{15}

Errors of approximation of optimal A_{16}

Errors of approximation of optimal A_{17}

Errors of approximation of optimal A_{18}

Figure 7: RBF errors in approximation of optimal cross-sectional areas.

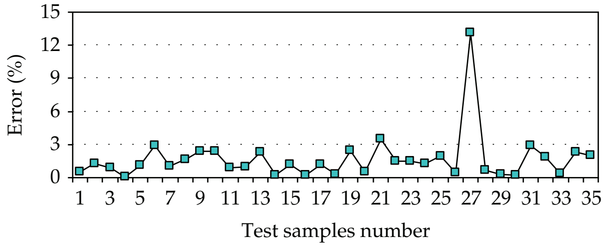


Figure 8: Errors of RBF for predicting the maximum deflections.

5.2. Main Steps in Training Neural Network

As a summary the main steps in training of RBF and GR NNs to predict optimal design and maximum deflection of the structure are as follows:

(1) configuration processing of the selected space structures employing Formian,

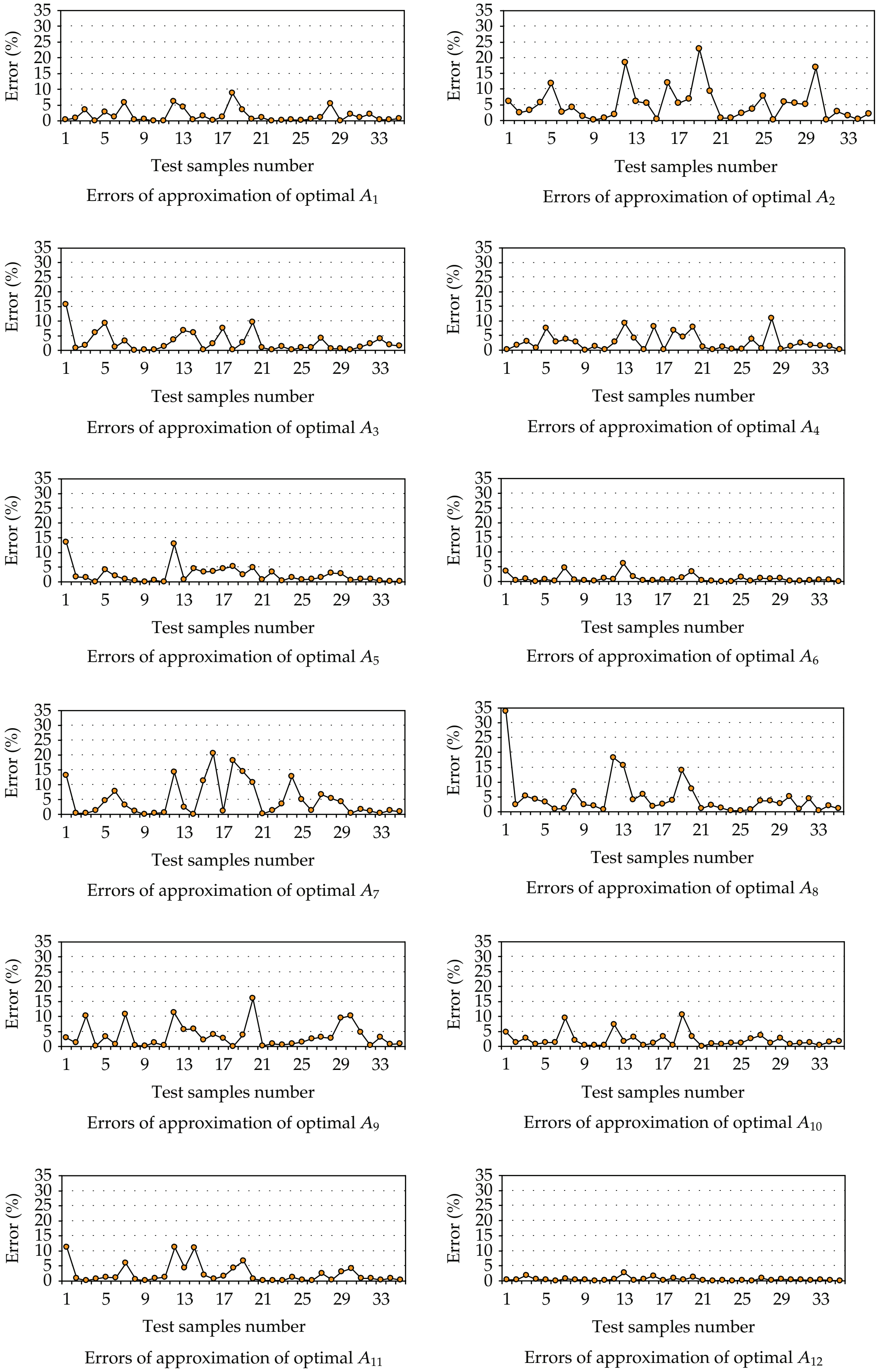
(2) selection a list of available tube sections from the standard lists,

(3) implementation member grouping,

(4) generation of some structures, based on span and height, to produce training set,

(5) static analysis of the structures,

(6) designing for optimal weight by SPSA according to AISC-ASD code,



Errors of approximation of optimal A_1

Errors of approximation of optimal A_2

Errors of approximation of optimal A_3

Errors of approximation of optimal A_4

Errors of approximation of optimal A_5

Errors of approximation of optimal A_6

Errors of approximation of optimal A_7

Errors of approximation of optimal A_8

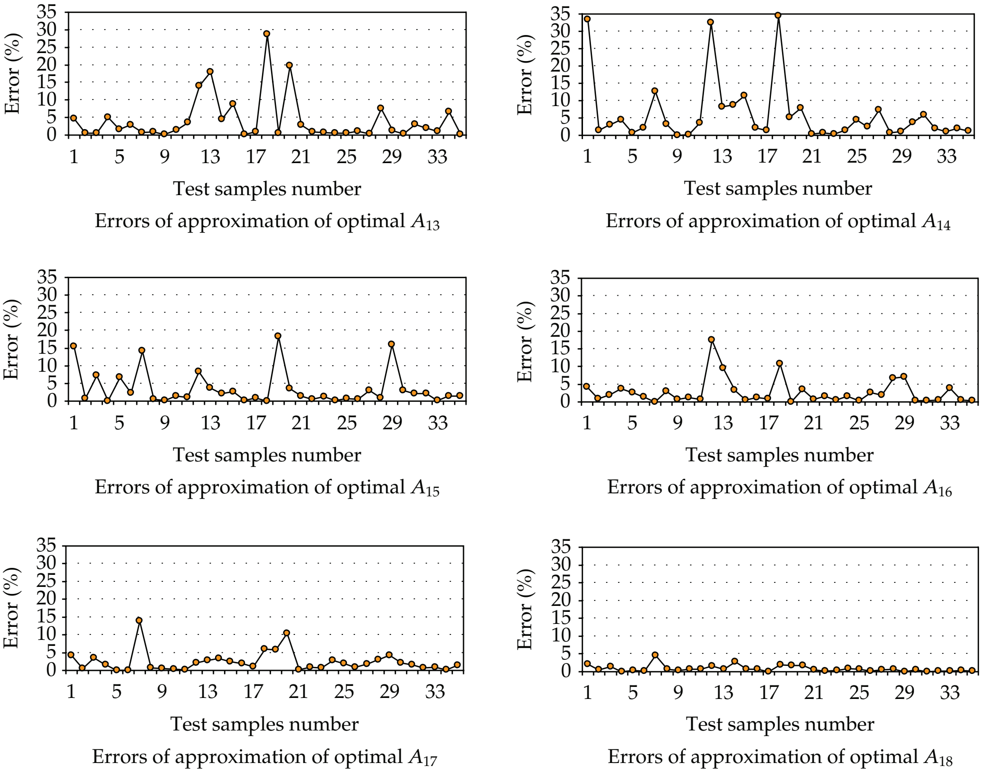
Errors of approximation of optimal A_9

Errors of approximation of optimal A_{10}

Errors of approximation of optimal A_{11}

Errors of approximation of optimal A_{12}

Figure 9: Continued.



Errors of approximation of optimal A_{13}

Errors of approximation of optimal A_{14}

Errors of approximation of optimal A_{15}

Errors of approximation of optimal A_{16}

Errors of approximation of optimal A_{17}

Errors of approximation of optimal A_{18}

Figure 9: GR errors in approximation of optimal cross-sectional areas.

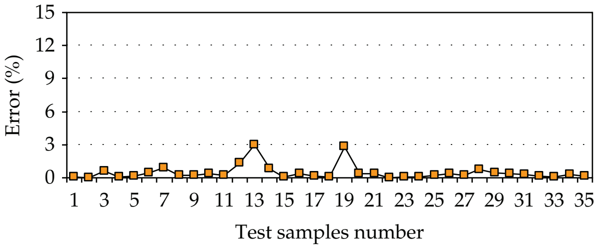


Figure 10: Errors of GR for predicting the maximum deflections.

(7) training and testing RBF and GR to predict optimal design and maximum deflection,

(8) improving generalization of the neural networks if it is necessary.

5.3. Flowchart of the Methodology

The flowchart of the proposed methodology is shown in Figure 4. This flowchart includes three main blocks: data generation, optimization, and NN training. The data generation block includes the optimization block. In these two blocks the data needed for neural network training is produced. The mentioned data are stated through (5.4) to (5.7).

Table 1: Summary of errors of RBF and GRNN in approximation of optimal designs.

Cross-sectional area	RBF		GRNN	
	Max. error	Mean of errors	Max. error	Mean of errors
1	32.8501	3.9054	8.8182	1.6575
2	30.0348	8.7955	22.8185	5.2312
3	18.3493	4.1153	15.6306	2.8343
4	36.8009	5.5074	10.8515	2.7220
5	15.6441	3.5933	13.4325	2.4476
6	24.8060	2.4974	6.1103	0.9992
7	33.3249	9.2145	20.4911	4.9154
8	18.7265	6.1936	33.8892	4.7607
9	26.1423	5.6690	16.1963	3.6055
10	29.7808	3.8814	10.6282	2.1702
11	21.9747	3.5231	11.3200	2.3858
12	29.3597	2.6233	2.6899	0.4795
13	26.4572	6.4117	28.6768	4.1939
14	23.7474	6.6549	37.5173	6.1356
15	29.8091	5.7633	18.2707	3.5529
16	32.8732	5.3975	17.5221	2.7931
17	21.5389	3.7532	13.8589	2.4312
18	29.6637	2.7991	4.4832	0.8361
Avr.	26.7714	5.0166	16.2892	3.0185

Table 2: Summary of errors of RBF and GRNN in approximation of maximum deflection.

RBF		GRNN	
Max. error	Mean of errors	Max. error	Mean of errors
13.1166	1.6675	3.0084	0.4641

6. Numerical Results

Typical topology of the RBF and GR neural networks to predict the optimal design and maximum deflection of the double layer grids is shown in Figures 5 and 6, respectively.

To find the optimal spread in the RBF and GR networks the minimum distance between training set and test set errors are employed [29]. The spread values in RBF networks trained to predict the optimal design and maximum deflection are 11.5 and 11.75 and for GR are 12.5 and 10.25, respectively. The results of RBF for predicting the optimal cross-sectional areas are shown in Figure 7.

The errors of RBF for predicting the maximum deflections are shown in Figure 8. The results of GR for predicting the optimal cross-sectional areas are shown in Figure 9. The errors of GR for predicting the maximum deflections are shown in Figure 10. Maximum and mean of errors of RBF and GRNN in approximation of optimal designs and maximum deflection are given in Tables 1 and 2, respectively.

The numerical results demonstrate that the generality of the GR is better than that of the RBF neural network in prediction of optimal design and maximum deflection of the double layer grids.

7. Conclusion

In this investigation, an innovative methodology is proposed to predict the optimal design and maximum deflection of the square-on-square double layer grids. This methodology consists of three stages. In the first stage, a number of the double layer grids with random spans and heights are generated. In the second stage the generated double layer grids are optimized by SPSA algorithm. Also, the maximum deflections of the optimal structures are saved. In the third stage, RBF and GR neural networks are trained to predict the optimal design and maximum deflection of the double layer grids.

By concerning the following points, it can be observed that the proposed methodology is novel and innovative.

(1) It is the first study based on employing the SPSA optimization algorithm to optimize double layer grids with variable geometry.

(2) Application of the RBF and GR neural networks to predict the optimal design and maximum deflection of the double layer is achieved for the first time in this study.

(3) The main advantage of the proposed methodology is to predict the optimal design and maximum deflection of the double layer grids with high speed and trivial errors in comparison with the traditional methods.

References

[1] M. L. Maher, D. Sriram, and S. J. Fenves, “Tools and techniques for knowledge based expert systems for engineering design,” *Advances in Engineering Software*, vol. 6, no. 4, pp. 178–188, 1984.

[2] R. Davis and J. J. King, “An overview of production systems,” in *Machine Intelligence 8: Machine Representations and Knowledge*, E. Elcock and D. Michie, Eds., pp. 300–332, John Wiley & Sons, New York, NY, USA, 1977.

[3] C. N. Kostem and M. L. Maher, Eds., *First Symposium on Expert Systems in Civil Engineering*, ASCE, Seattle, Wash, USA, 1986.

[4] C. E. Shannon, “Programming a computer for playing chess,” *Philosophical Magazine*, vol. 41, series 7, no. 314, pp. 256–275, 1950.

[5] F. Erbatur, O. Hasançebi, I. Tütüncü, and H. Kinodot;liç, “Optimal design of planar and space structures with genetic algorithms,” *Computers and Structures*, vol. 75, no. 2, pp. 209–224, 2000.

[6] S. Rajasekaran, “Optimization of large scale three dimensional reticulated structures using cellular genetics and neural networks,” *International Journal of Space Structures*, vol. 16, no. 4, pp. 315–324, 2001.

[7] C. S. Krishnamoorthy, V. P. Prasanna, and R. Sudarshan, “Object-oriented framework for genetic algorithms with application to space truss optimization,” *Journal of Computing in Civil Engineering*, vol. 16, no. 1, pp. 66–75, 2002.

[8] A. Tashakori and H. Adeli, “Optimum design of cold-formed steel space structures using neural dynamics model,” *Journal of Constructional Steel Research*, vol. 58, no. 12, pp. 1545–1566, 2002.

[9] E. Salajegheh and J. Salajegheh, “Continuous-discrete variable optimization of space structures with sizing variables using approximation methods,” *International Journal of Space Structures*, vol. 16, no. 4, pp. 237–251, 2001.

[10] A. Kaveh and M. R. Dehkordi, “Neural networks for the analysis and design of domes,” *International Journal of Space Structures*, vol. 18, no. 3, pp. 181–193, 2003.

[11] A. Kaveh, Y. Gholipour, and H. Rahami, “Optimal design of transmission towers using genetic algorithm and neural networks,” *International Journal of Space Structures*, vol. 23, no. 1, pp. 1–19, 2008.

[12] A. Kaveh and H. Servati, “Design of double layer grids using backpropagation neural networks,” *Computers and Structures*, vol. 79, no. 17, pp. 1561–1568, 2001.

[13] E. Salajegheh and S. Gholizadeh, “Optimum design of structures by an improved genetic algorithm using neural networks,” *Advances in Engineering Software*, vol. 36, no. 11-12, pp. 757–767, 2005.

[14] A. Kaveh, B. F. Azar, and S. Talatahari, “Ant colony optimization for design of space trusses,” *International Journal of Space Structures*, vol. 23, no. 3, pp. 167–181, 2008.

[15] S. Gholizadeh, E. Salajegheh, and P. Torkzadeh, “Structural optimization with frequency constraints by genetic algorithm using wavelet radial basis function neural network,” *Journal of Sound and Vibration*, vol. 312, no. 1-2, pp. 316–331, 2008.

[16] S. Gholizadeh and E. Salajegheh, “Optimal design of structures subjected to time history loading by swarm intelligence and an advanced metamodel,” *Computer Methods in Applied Mechanics and Engineering*, vol. 198, no. 37–40, pp. 2936–2949, 2009.

[17] A. H. Alavi and A. H. Gandomi, “Prediction of principal ground-motion parameters using a hybrid method coupling artificial neural networks and simulated annealing,” *Computers and Structures*, vol. 89, no. 23-24, pp. 2176–2194, 2011.

[18] S. Gholizadeh, A. Pirmoz, and R. Attarnejad, “Assessment of load carrying capacity of castellated steel beams by neural networks,” *Journal of Constructional Steel Research*, vol. 67, no. 5, pp. 770–779, 2011.

[19] A. H. Gandomi and A. H. Alavi, “Applications of computational intelligence in behavior simulation of concrete materials,” in *Computational Optimization and Applications in Engineering and Industry*, X. S. Yang and S. Koziel, Eds., vol. 359, chapter 9, pp. 221–243, Springer, New York, NY, USA, 2011.

[20] S. Gholizadeh and S. M. Seyedpoor, “Shape optimization of arch dams by metaheuristics and neural networks for frequency constraints,” *Scientia Iranica*, vol. 18, no. 5, pp. 1020–1027, 2011.

[21] S. Talatahari, A. H. Gandomi, and G. J. Yun, “Optimum design of tower structures using firefly algorithsm,” *The Structural Design of Tall and Special Buildings*. In press.

[22] S. Gholizadeh and A. Barzegar, “Shape optimization of structures for frequency constraints by sequential harmony search algorithm,” *Engineering Optimization*. In press.

[23] A. H. Gandomi, S. Talatahari, X. S. Yang, and S. Deb, “Design optimization of truss structures using cuckoo search algorithm,” *The Structural Design of Tall and Special Buildings*. In press.

[24] S. Gholizadeh and F. Fattahi, “Design optimization of tall steel buildings by a modified particle swarm algorithm,” *The Structural Design of Tall and Special Buildings*. In press.

[25] S. Talatahari, M. Kheirollahi, C. Farahmandpour, and A. H. Gandomi, “A multi-stage particleswarm for optimum design of truss structures,” *Neural Computing & Applications*. In press.

[26] J. C. Spall, “An overview of the simultaneous perturbation method for efficient optimization,” *Johns Hopkins APL Technical Digest*, vol. 19, no. 4, pp. 482–492, 1998.

[27] P. D. Wasserman, *Advanced Methods in Neural Computing*, Prentice Hall, D. Van Nostrand, New York, NY, USA, 1993.

[28] The Language of Technical Computing, *MATLAB*, Math Works, 2006.

[29] S. S. Sadat Hosseini and A. H. Gandomi, “Short-term load forecasting of power systems by gene expression programming,” *Neural Computing and Applications*, vol. 21, no. 2, pp. 377–389, 2012.