ASIAN JOURNAL OF CIVIL ENGINEERING (BUILDING AND HOUSING) VOL. 11, NO. 1 (2010)
PAGES 25-42

STRUCTURAL OPTIMIZATION FOR EARTHQUAKE LOADING WITH NONLINEAR RESPONSES BY SURROGATE MODELING BASED EVOLUTIONARY ALGORITHMS

S. Gholizadeh[*]
Department of Civil Engineering, Urmia University, Urmia, Iran

ABSTRACT

An efficient methodology is proposed to design optimization of structures subjected to earthquake time history loading considering nonlinear structural response. It is clear that the structural optimization for transient time history loading is a computationally intensive task, especially when the nonlinear response is concerned. In the proposed hybrid methodology particle swarm optimization (PSO), genetic algorithm (GA), probabilistic neural network (PNN), radial basis function neural network (RBFNN) and wavelet transforms (WT) techniques are combined to achieve the optimization task. In order to investigate the efficiency of the proposed methodology, a 72-bar space steel tower is designed for optimal weight for El Centro earthquake. The numerical results demonstrate the efficiency and computational advantages of the proposed methodology.

Keywords: Optimization; earthquake; genetic algorithm; particle swarm optimization; neural network

1. INTRODUCTION

Seismic design codes suggest that, under severe earthquake events, the structures should be designed to deform inelastically due to the large intensity inertia loads imposed. Therefore, achieving the structural optimization for transient time history loading, considering nonlinear response is an expensive process in terms of computational costs. In order to deal with this problem an efficient methodology is proposed. In the proposed methodology two main computational strategies have been adopted. In the first strategy, particle swarm optimization (PSO) algorithm [1] and genetic algorithm (GA) [2] are employed to achieve the optimization task. In the field of structural optimization, many successful application of the GA, PSO and their modified versions have been reported in literature [3-8]. As the optimization process requires a great number of nonlinear time history analyses thus the overall time of the optimization process is very long. In the second strategy, in order to reduce the computational burden, a surrogate model based on radial basis function (RBF)

[*] Email-address of the corresponding author: s.gholizadeh@urmia.ac.ir (S. Gholizadeh)

[9] neural network and wavelet transforms (WT) [10] is employed to predict the necessary responses during the optimization process. One of the most important phases in the surrogate modeling is sampling (data generation). In this paper, the optimal Latin hypercube (OLH) [11] sampling technique is employed to select efficient data set. For prediction of the responses, a wavelet radial basis function neural network (WRBFNN) is employed. There are two alternatives for the input vectors of the WRBFNN: design variables and natural frequencies. In [12] it has been demonstrated that employing the natural frequencies as the inputs of surrogates results in better performance generality. As the natural frequencies are required during the optimization process, evaluating of these by analytic methods can impose additional computational burden to the process. In order to prevent from this difficulty, the WRBFNN proposed in [13] is employed to effectively predict the frequencies. During the optimization process of a structure many designs are examined and due to considering nonlinear response analysis it is probable that some of them lose their stability. It is evident that such designs should be rejected. As during the optimization process, we have employed a surrogate model instead of the exact nonlinear time history analysis, it is necessary to somehow distinguish such instable structures. For this purpose, a probabilistic neural network (PNN) [14] has been trained. As a test example, a 72-bar space steel tower subjected to the El Centro earthquake is optimized. The numerical results indicate that the hybrid methodology is a powerful and efficient tool for optimal design of structures.

2. FORMULATION OF THE OPTIMAL DESIGN PROBLEM FOR EARTHQUAKE LOADING

Optimal design of structures subjected to earthquake time history loading can be formulated as follows:

$$\text{Find } \boldsymbol{X};\ X_i \in R^d;\ i = 1,\cdots,n \tag{1}$$

$$\text{To Minimize } W(\boldsymbol{X}) \tag{2}$$

$$\text{Subject to } g_j(\boldsymbol{X}, \boldsymbol{Z}(t), \dot{\boldsymbol{Z}}(t), \ddot{\boldsymbol{Z}}(t), t) \le 0;\ j = 1,\cdots,m \tag{3}$$

$$\boldsymbol{M}\ddot{\boldsymbol{Z}}(t) + \boldsymbol{C}\dot{\boldsymbol{Z}}(t) + \boldsymbol{K}\boldsymbol{Z}(t) - \boldsymbol{M}\boldsymbol{I}\,\ddot{u}_g(t) = 0 \tag{4}$$

where W, g, $\boldsymbol{I}$, $\boldsymbol{X}$, $\ddot{\boldsymbol{Z}}(t)$, $\dot{\boldsymbol{Z}}(t)$, $\boldsymbol{Z}(t)$, $\boldsymbol{M}$, $\boldsymbol{C}$, $\boldsymbol{K}$, $\ddot{u}_g(t)$,m, n, and t are objective function, behavioral constraint, unit vector, design variables vector, acceleration vector, velocity vector, displacement vector, mass matrix, damping matrix, stiffness matrix, ground acceleration, the number of constraints, the number of design variables, and time, respectively. R^d is a given set of discrete values.

In order to perform dynamic analysis considering inelastic behavior, ANSYS [15] is employed. A simple full plastic stress-strain relationship is adopted in order to take into account the transient nature of earthquake loading. Studies have shown that this law is adequate and gives accurate results for many practical applications [16]. The full plastic stress-strain relationship is shown in Figure 1.

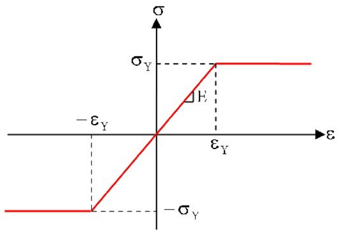

Figure 1. Full plastic stress-strain relationship

The stress and displacement constraints in time-dependent optimization problems are usually taken as:

$$\text{if } \frac{f_{ai}(X,t)}{F_a} \le 0.15: \quad g_{qi}(X,t) = \frac{f_{ai}(X,t)}{F_a} + \frac{f_{byi}(X,t)}{F_{by}} + \frac{f_{bzi}(X,t)}{F_{bz}} - 1 \le 0 \; ; \; i = 1,\ldots,ne \tag{5}$$

$$\text{if } \frac{f_{ai}(X,t)}{F_a} > 0.15: \begin{cases} g_{qi}(X,t) = \dfrac{f_{ai}(X,t)}{F_a} + \dfrac{C_{mx} f_{byi}(X,t)}{(1 - f_{ai}(X,t)/F'_{ey})F_{by}} + \dfrac{C_{my} f_{bzi}(X,t)}{(1 - f_{ai}(X,t)/F'_{ez})F_{bz}} - 1 \le 0 \\ g_{qi}(X,t) = \dfrac{f_{ai}(X,t)}{0.6F_y} + \dfrac{f_{byi}(X,t)}{F_{by}} + \dfrac{f_{bzi}(X,t)}{F_{bz}} - 1 \le 0 \end{cases} ; i = 1,\ldots,ne \tag{6}$$

$$g_{dj}(X,t) = \frac{d_j(X,t)}{d_{j,all}} - 1 \le 0 ; \; j = 1,\ldots,ns \tag{7}$$

where *ne*, *ns*, d_j, f_{ai}, f_{byi} and f_{bzi} are the number of the elements, the number of the nodes, drift of the *j*th storey, compressive axial stress, bending stress for y-axis, bending stress for z-axis of *i*th element, respectively; $d_{j,all}$, F_a, F_{by} and F_{bz} are allowable values of the mentioned parameters, respectively; F_y is the yield stress. Also C_{mx} and C_{my} are bending coefficients, and F'_{ey} and F'_{ez} are the Euler stresses.

As all the constraints are time-dependent the consideration of all the constraints requires an enormous amount of computational effort. In the present study, the conventional method [17] is employed to deal with time-dependent constraints. In this method the time interval is divided into n_{gp} subintervals and the time-dependent constraints are imposed at each time grid point. Let the *j*th time-dependent constraint be written as:

$$g_j(\mathbf{X}, \mathbf{Z}(t), \dot{\mathbf{Z}}(t), \ddot{\mathbf{Z}}(t), t) \le 0, \quad 0 \le t \le ti \tag{8}$$

where *ti* is time interval over which the constraints need to be imposed.

Because the total time interval is divided into n_{gp} subintervals, the constraint (8) is replaced by the constraints at the n_{gp}+1 time grid points as:

$$g_j(\boldsymbol{X}, \boldsymbol{Z}(t_\alpha), \dot{\boldsymbol{Z}}(t_\alpha), \ddot{\boldsymbol{Z}}(t_\alpha), t_\alpha) \le 0; \; \alpha = 0, \cdots, n_{gp} \tag{9}$$

The above constraint function can be evaluated at each time grid point after the structure has been analyzed.

The objective function of constrained structural optimization problems is defined as follows:

$$W(\boldsymbol{X}) = \begin{cases} W(\boldsymbol{X}) & \text{if } \boldsymbol{X} \in \tilde{\Delta} \\ W(\boldsymbol{X}) + f_p(\boldsymbol{X}) & \text{otherwise} \end{cases}, \quad f_p(\boldsymbol{X}) = \sum_{\alpha=0}^{n_{gp}} r_P \left[\sum_{j=1}^{m} \left(\max(g_j(\boldsymbol{X}, t_\alpha), 0) \right)^2 \right] \tag{10}$$

where $f_p(\boldsymbol{X})$, $\tilde{\Delta}$, and r_p are the penalty function, the feasible search space, and an adjusting factor, respectively.

3. EVOLUTIONARY ALGORITHMS

In the present study, the GA and PSO are employed for performing optimization process. During the last years, the GA has been widely employed to solve the engineering optimization problems and its theoretical background has been described in the literature. Thus in this paper, only the theoretical background of the PSO is briefly described.

3.1 PSO Algorithm

The PSO has been proposed by Kennedy [1] to simulate the graceful motion of bird swarms as a part of a socio-cognitive study. The PSO involves a number of particles, which are randomly initialized in the search space. These particles are referred to as swarm. Each particle of the swarm represents a potential solution of the optimization problem. The particles fly through the search space and their positions are updated based on the best positions of individual particles and the best of the swarm in each iteration. The objective function is evaluated for each particle at each grid point and the fitness values of particles are obtained to determine the best position in the search space [18]. In iteration *k*, the swarm is updated using the following equations:

$$V_i^{k+1} = \omega^k V_i^k + c_1 r_1 (P_i^k - X_i^k) + c_2 r_2 (P_g^k - X_i^k) \tag{11}$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \tag{12}$$

where X_i and V_i represent the current position and the velocity of the *i*th particle, respectively; P_i is the best previous position of the *i*th particle (called *pbest*) and P_g is the best global position among all the particles in the swarm (called *gbest*); r_1 and r_2 are two uniform random sequences generated from interval [0, 1]; c_1 and c_2 are the cognitive and social scaling parameters, respectively. Each component of V_i is constrained to a maximum value defined as V_i^{max} and a minimum value defined as V_i^{min}. The inertia weight used to discount the previous velocity of particle preserved is expressed by ω.

Due to the importance of ω in achieving efficient search behavior the optimal updating criterion is taken as:

$$\omega = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{k_{max}} . k \qquad (13)$$

where ω_{max} and $\omega_{m\text{in}}$ are the maximum and minimum values of ω, respectively. Also, k_{max}, and k are the numbers of maximum iterations and present iteration, respectively.

It was demonstrated that $c_1 = c_2 = 2.0$ [19] therefore in the present paper it is assumed that $c_1 = c_2 = 2.0$ and proper results have been observed. The particle size is a problem dependent parameter. However, in [20] it is mentioned that the typical range for the number of particles is 20–40. In this work, a particle size of 30 is chosen and proper results are found.

The flowchart of the typical PSO is shown in Figure 2.

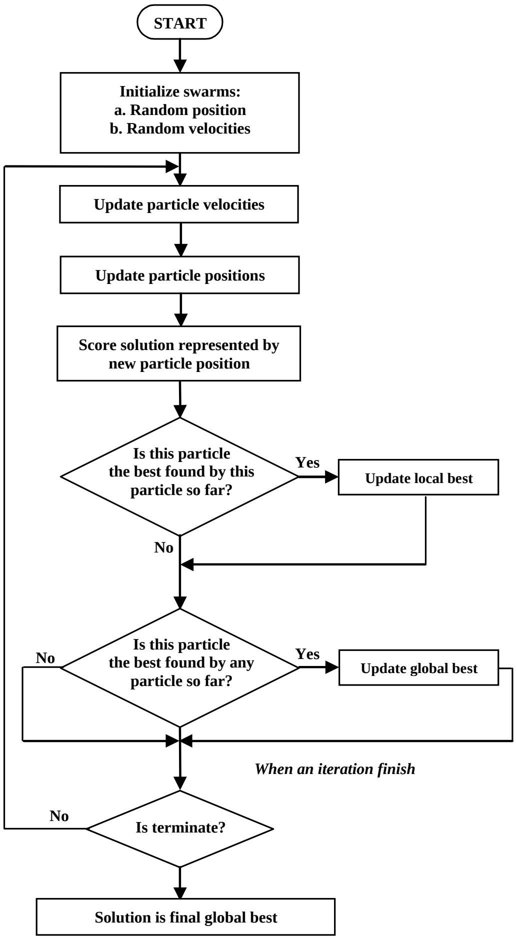
START
Initialize swarms:
a. Random position
b. Random velocities
Update particle velocities
Update particle positions
Score solution represented by new particle position
Is this particle the best found by this particle so far?
Yes
Update local best
No
Is this particle the best found by any particle so far?
No
Yes
Update global best
When an iteration finish
No
Is terminate?
Solution is final global best

Figure 2. The flowchart of the typical PSO

Despite the efficiency of the PSO, the computational burden of the optimization process due to implementing nonlinear time history analyses is very high. Incorporating of surrogate models in the optimization process to predict the desired structural response can substantially reduce the computing effort.

4. SURROGATE MODELING FOR PREDICTION OF THE NONLINEAR TIME HISTORY RESPONSES

In this study a hybrid surrogate model (HSM) is proposed to efficiently and accurately predict the nonlinear time history responses of the structures. In this model, OLH, WRBFNNs and PNN are combined. In the next sub-sections, a brief description of theoretical background of the mentioned techniques is presented and then the fundamentals of the HSM are explained.

4.1 OLH Sampling Method

Design of computer experiments [21] is often used for response database construction, by generating a set of representative input vectors that should cover the entire design space and be spread as evenly as possible. Design of the experiments is usually time-consuming and only a limited number of them can be performed. Therefore, a careful selection of the data set is of primary importance. Latin hypercube design [11] of experiments and the so-called optimal Latin hypercube (OLH) provide good alternative to this purpose. However, OLHs are rather expensive to employ for high dimensional problems.

A Latin hypercube is represented by an $N_S \times p$ matrix $\boldsymbol{L}$ in which each column consists of a permutation of the integers 1 to N_S. Each row of $\boldsymbol{L}$ is considered as a sample point in p dimensions.

$$\boldsymbol{L} = \begin{bmatrix} \boldsymbol{X}_1 \\ \vdots \\ \boldsymbol{X}_{N_S} \end{bmatrix} = \begin{bmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{N_S 1} & \cdots & x_{N_S p} \end{bmatrix} \tag{14}$$

where $\boldsymbol{X}_i$, $1 \le i \le N_S$, is the ith sample point.

In the OLH approach, the objective is an optimized Latin hypercube design so that the neighboring data points are kept at a minimal distance apart. The criterion that is used in the present paper to optimize LH design was proposed by Audze and Eglais [22]. It is based on the function G given below:

$$\mathrm{G}(\boldsymbol{L}) = \sum_{i=1}^{N_S} \sum_{j=i+1}^{N_S} \frac{1}{(\boldsymbol{X}_i \text{ - } \boldsymbol{X}_j)^2} \tag{15}$$

The optimization problem consists of finding an LH (represented by $\boldsymbol{L}$) which minimizes G($\boldsymbol{L}$). For this purpose the columnwise-pairwise (CP) algorithm [23] is employed.

4.2 WRBFNN

Researchers have proven that the wavelet type of neural networks possess better performance generality in comparison with their conventional versions [24, 25]. A wavelet neural network logically connects neural network architecture with the wavelet transform. A wavelet family associated with mother wavelet $\psi(z)$ is generated by dilation (a > 0) and translation (b) factors:

$$\psi_{a,b}(z)=\frac{1}{\sqrt{a}}\psi(\frac{z-b}{a}) \tag{16}$$

The wavelet neural networks use wavelets as activation functions of hidden layer neurons. In this paper, the dilation and translation factors of the wavelet functions are taken to be fixed and only the weights of the network are determined. Activation function of the conventional RBF neurons is Gaussian function. To design the WRBFNN the activation function of hidden layer of RBFNN is substituted with Morlet [26] wavelet function:

$$\psi^{Morlet}(z)=\cos(4z)\exp(-0.5z^2) \tag{17}$$

$$\psi_{a,b}^{Morlet}(z)=\frac{1}{\sqrt{a}}\cos(4(\frac{\text{z - }b}{a}))\exp(-0.5(\frac{z-b}{a})^2) \tag{18}$$

In order to find the optimal values of a and b a simple approach is adopted. In this approach, the values of a and b are iteratively changed and the performance generality of the FWRBFNN is investigated. The values associated with the best performance generality are taken as optimal. Due to the high training speed of the RBF type networks, the approach is very fast [13].

4.3 PNN

The probabilistic neural network (PNN) proposed by Specht (1990) is mainly used for classification problems. To train the PNN a supervised training is accomplished. Typically, a PNN consists of an input layer, a radial basis function (RBF) layer, and a competitive (C) layer.

During the training stage, a training set of N_S data samples (design variable vectors) is used.

$$\boldsymbol{I}_{\text{PNN}}=[\boldsymbol{F}_1,\boldsymbol{F}_2,...,\boldsymbol{F}_j,...,\boldsymbol{F}_{N_S}]\ ,\boldsymbol{F}_j=\{f_1^j,f_2^j,...,f_q^j\}^{\text{T}},\ j=1,2,...,N_S \tag{19}$$

where q is the number of the elements of each sample.

The number of the neurons in the RBF layer is identical to N_S. Also, the weight matrix of this layer is set to the transpose of the $\boldsymbol{I}_{\text{PNN}}$ matrix [27].

The number of the neurons in the competitive layer, which is identical to the count of the classes, is denoted by N_C. The weight of the kth neuron in the competitive layer to the jth neuron in the RBF layer is assigned as

$$w_{k,j}^{\text{C}}=\begin{cases}1 & \text{if}\boldsymbol{F}_j\in\boldsymbol{C}_k\\ 0 & \text{otherwise}\end{cases} \tag{20}$$

where $\boldsymbol{C}_k$ denotes the sample set of the kth class.

Based on these assignments, the PNN is created with zero error on training samples.

After training, a testing set of N_T new data samples are used to test the performance generality of the PNN. When a new input vector $\boldsymbol{F}_* = \{f_1^*, f_2^*, ..., f_q^*\}^{\mathrm{T}}$ is presented, the RBF layer first computes the distances between $\boldsymbol{F}_*$ and the training samples. Then RBF activation function is used to produce a vector whose elements indicate how close the input vector is to the training sample. The output of the jth activation function in the RBF layer is

$$\varphi_j = \exp(-\frac{\sum_{i=1}^{m} (w_{j,i}^{\mathrm{RBF}} - x_i^*)^2}{2\sigma_j^2}), \quad j = 1,2,...,N_S \tag{21}$$

where σ_j is the radius of receptive field of the jth RBF neuron [27].

Thus, the RBF layer neurons with weight vectors quite far from the input vector $\boldsymbol{F}_*$ output values near zero, while neurons with weight vectors quite close to $\boldsymbol{F}_*$ output values near one. Typically, several neurons may be active to varying degrees. The competitive layer sums these contributions for each class of inputs to produce a vector of probabilities. Each neuron in the competitive layer represents the active status of one class as follows:

$$\xi_k = \frac{\sum_{j=1}^{N_S} w_{k,j}^{\mathrm{C}} \varphi_j}{\sum_{j=1}^{N_S} w_{k,j}^{\mathrm{C}}}, \quad k = 1,2,...,N_C \tag{22}$$

Finally, the second layer, produces a 1 corresponding to the largest element of ξ_k, ($k = 1,2,...,N_C$), and 0's elsewhere. Thus, the network classifies the input vector into a specific k ($k = 1,2,...,N_C$) class because that class has the maximum probability of being correct.

The key advantage of PNN over the other networks is its rapid training. Since the number of layers in the PNN architecture is fixed and all the synaptic weights are directly assigned using training samples, this procedure can be finished in only one epoch and no error correction procedure is necessary. It has been proved that with enough training data a PNN is guaranteed to converge to a Bayesian classifier, which usually owns the optimal classification capability [27].

4.4 Fundamentals of HSM

Predicting the linear time history responses of the structures have been already achieved in a number of papers [28-30]. In this paper, a HSM is proposed to predict the nonlinear time history responses of structures.

As the first step, the OLH sampling method is employed to carefully select a data set. In the sampling process, N_S structures based on their design variable vectors are selected. This means that in Eq.(14), $\boldsymbol{X}_i$ is the design variables of the ith selected structure. In this case, the natural frequencies ($\boldsymbol{F}_i$) and nonlinear time history responses ($\boldsymbol{R}_i$) of all the selected structures are computed by the conventional finite elements (FE) analysis.

In [12] it has been demonstrated that the best candidates for the inputs of surrogates are natural frequencies. In the present paper, the natural frequencies are also employed as the inputs. During the optimization process evaluating of the frequencies by employing the analytic methods increases the computational effort of the process. In order to prevent from this, the WRBFNN [13] is used to predict the frequencies. The inputs and outputs of the WRBFNN are design variables ($\boldsymbol{X}_i$) and natural frequencies ($\boldsymbol{F}_i$) of the selected structures, respectively.

During the nonlinear time history analysis of a structure due to updating the structural matrices, it is probable that the structure loses its overall stability and the analysis procedure can not converge. Thus, before training a surrogate model to predict the nonlinear responses, it is crucial to recognize stable and instable structures. In order to attain this purpose, classifier neural networks can be employed. In the present work, a PNN is trained to achieve this important task. All of the selected structures, N_S, are considered in the classification phase. Naturally, considering stable and instable structures the number of classes, N_C, is equal to 2. In this paper we examined two alternatives for the inputs of PNN: design variables ($\boldsymbol{X}_i$) and natural frequencies ($\boldsymbol{F}_i$). It is observed that employing $\boldsymbol{F}_i$ as the inputs results in appropriate results. Therefore, in the training process of the PNN, the inputs are $\boldsymbol{F}_i$ and the output is 1 for stability and 2 for instability of the corresponding structure. Employing N_{S1} stable and N_{S2} instable structures, the PNN is trained to recognize stable and instable structures during the optimization process.

The last stage in training the HSM is to train a network to predict the nonlinear time history responses of the N_{S1} stable structures. For this purpose, another WRBFNN is considered. The inputs and outputs of the WRBFF are $\boldsymbol{F}_i$ and $\boldsymbol{R}_i$ of the N_{S1} stable structures. As well as the frequency predictor WRBFNN, in this network the mentioned simple procedure is also employed to find optimal values of a and b.

4.5 Evaluation Metrics

In order to evaluate the accuracy of the approximate nonlinear time history responses (predicted by HSM) against their corresponding actual ones (obtained by conventional FE analysis), two evaluation metrics are used as:

$$\mathrm{RRMSE} = ((\frac{1}{r\text{-}1}\sum_{i=1}^{r}(z_i - \tilde{z}_i)^2)/(\frac{1}{r}\sum_{i=1}^{r}(z_i)^2))^{\frac{1}{2}} \text{ , } \mathrm{R}^2 = 1 - (\sum_{i=1}^{r}(z_i - \tilde{z}_i)^2)/(\sum_{i=1}^{r}(z_i - \bar{z})^2) \tag{23}$$

where z_i and $\tilde{z}_i$ are the ith component of the exact and approximate responses, respectively. The mean value of exact vectors component and the vectors dimension are expressed by $\bar{z}$ and r, respectively.

5. OPTIMIZATION BY THE PSO INCORPORATING THE HSM

The outline of the proposed methodology in this paper is displayed in Figure 3.

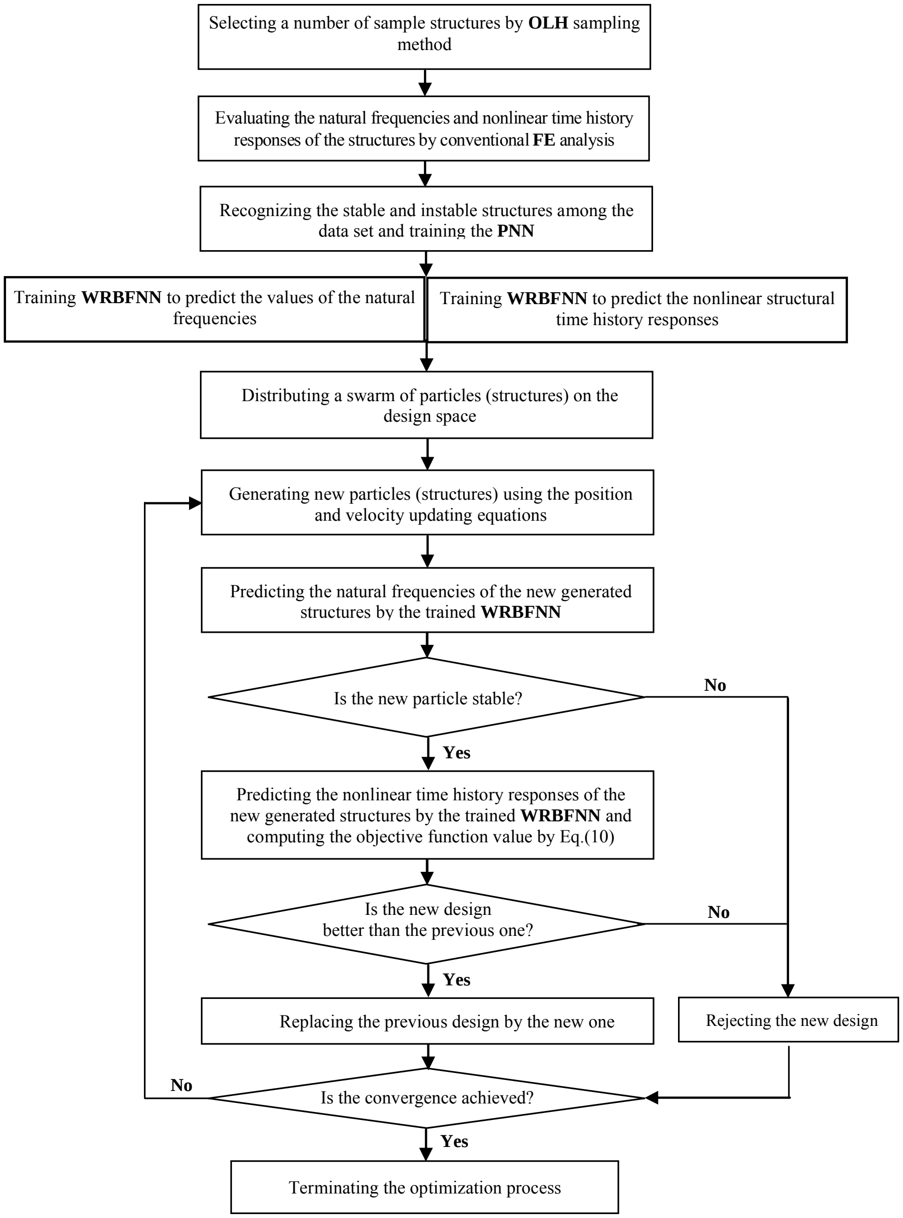
Selecting a number of sample structures by **OLH** sampling method
Evaluating the natural frequencies and nonlinear time history responses of the structures by conventional **FE** analysis
Recognizing the stable and instable structures among the data set and training the **PNN**
Training **WRBFNN** to predict the values of the natural frequencies
Training **WRBFNN** to predict the nonlinear structural time history responses
Distributing a swarm of particles (structures) on the design space
Generating new particles (structures) using the position and velocity updating equations
Predicting the natural frequencies of the new generated structures by the trained **WRBFNN**
Is the new particle stable?
No
Yes
Predicting the nonlinear time history responses of the new generated structures by the trained **WRBFNN** and computing the objective function value by Eq.(10)
Is the new design better than the previous one?
No
Yes
Replacing the previous design by the new one
Rejecting the new design
No
Is the convergence achieved?
Yes
Terminating the optimization process

Figure 3. Outline of the proposed methodology

6. NUMERICAL RESULTS

A 72-bar truss, shown in Figure 4, subjected to 10 seconds of the El Centro (S-E 1940) earthquake record is designed for optimal weight. The earthquake records are applied in x direction. Young's modulus is 2.1×10^{10} kg/m^2 and mass density is 7850 kg/m^3. The mass of 10000kg is lumped at nodes of 1 to 4. The truss members are divided into 9 groups based on cross-sectional areas, given in Table 1. The cross-sectional areas of the elements can be chosen from the values given in Table 2. Because of the insignificant internal stresses of elements of group 9 under the earthquake excitation, a minimum cross-sectional area of 2.54 cm^2 is assigned to them. This structure has been optimized considering linear elastic structural behavior in [28]. As in this paper, nonlinear structural behavior is considered the stress limitation is neglected and as well as the [28] the displacement of top node of the structure is limited to 2cm. The constraints are checked at 500 grid points with time step of 0.02 seconds. The computational time is measured in terms of CPU time required by a PC Pentium IV 3000 MHz.

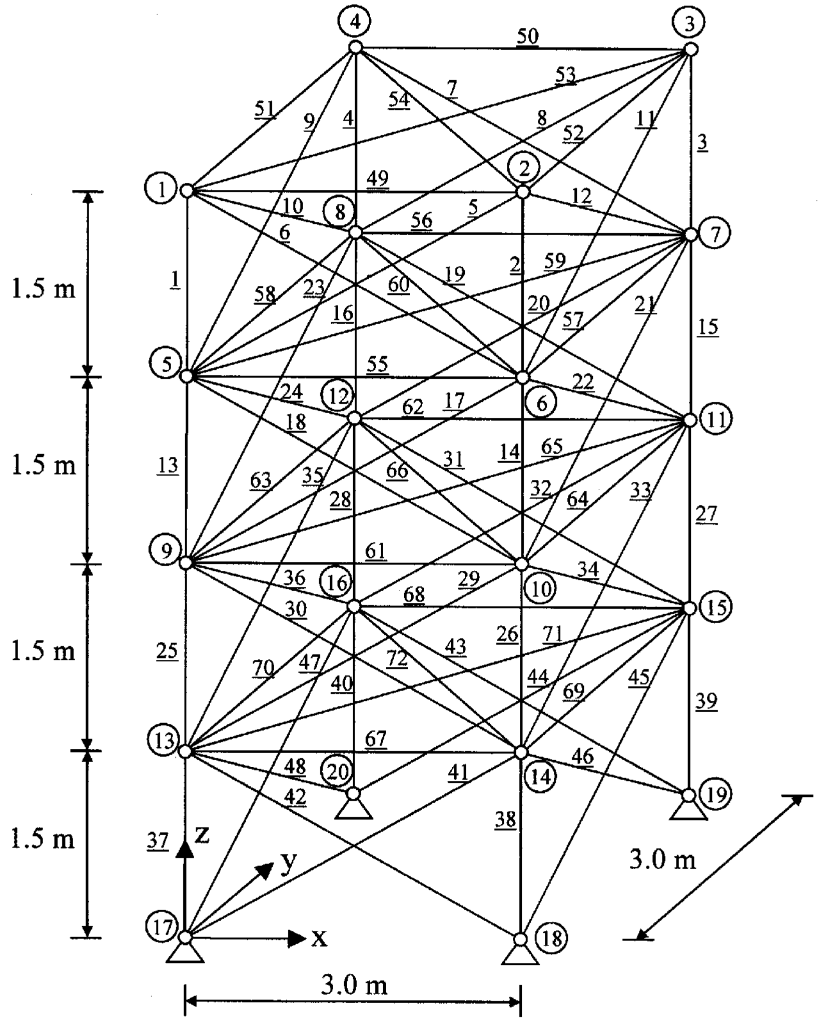

Figure 4. 72-Bar space steel truss

Table 1. Element groups of the 72-bar truss

Group No.	Elements
1	1-4
2	5-12
3	13-16
4	17-24
5	25-28
6	29-36
7	37-40
8	41-48
9	49-72

Table 2. Available cross-sectional areas

No.	Area (cm^2)
1	2.54
2	11.20
3	12.30
4	13.90
5	15.20
6	17.20
7	18.90
8	21.40
9	25.70

6.1 Data selection

To train and test the HSM a training set including 165 samples are selected by OLH sampling technique, N_S=150. The time spent to achieve OLH sampling process is about 5.0min.

Natural frequencies and nonlinear time history responses of the selected structures are evaluated by the conventional FE analysis. During this process it is revealed that among all the structures, 15 ones lose their stability and their analysis procedure can not converge. Therefore, in the data set we have N_{S1}=150 stable and N_{S2}=15 instable structures. The time spent to FE analysis of 165 structures is 235 min.

6.2 Training WRBFNN to Predict the Natural Frequencies

A WRBFNN is trained to predict the natural frequencies of the structures during the optimization process. The inputs and outputs of the WRBFNN are design variables ($\boldsymbol{X}_i$, i=1,2,…, 165) and frequencies ($\boldsymbol{F}_i$, i=1,2,…,165) of the selected structures, respectively. Due to symmetry of the structure, its 1st, 3rd and 5th natural frequencies are selected as inputs. From the 165 selected structures, 110 and 55 ones are randomly selected to train and test the WRBFNN, respectively. In this case, the size of the WRBFNN is 9-110-3. In this example, the optimal values of *a* and *b* are obtained as 0.8 and 0.0, respectively. The results of testing the performance generality of the WRBFNN are given in Table 3. The time spent to train and test the WRBFNN is 1.5min.

Table 3. Testing errors of the WRBFNN

Natural frequencies	**Mean error (%)**	**Maximum error (%)**
f_1	0.3814	1.3133
f_3	0.2100	1.0008
f_5	0.2926	1.4548
Average	0.2947	1.2563

6.3 Training PNN to Recognize Stable and Instable Structures

In the provided data set, there are N_{S1}=150 stable and N_{S2}=15 instable structures. In order to train the PNN, 110 samples including 100 stable and 10 instable structures are considered. Also to test the PNN, 55 samples including 50 and 5 stable and instable ones are considered, respectively. The testing results of PNN are shown in Figure 5. It can be observed that there is excellent conformance between exact results and predicted ones by PNN. The time spent to train and test the PNN is 1.0min.

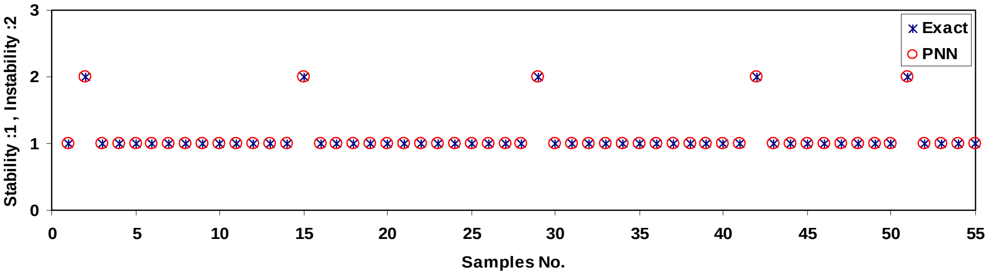

Figure 5. Testing results of PNN

6.4 Training WRBFNN to Predict the Nonlinear Time history Responses of the Stable Structures

A WRBFNN is trained to predict the displacement of the top node of the structure. As the instable structures will be rejected during the optimization process, therefore the WRBFNN is only trained to predict the nonlinear responses of the stable structures. In this case, the training set includes the N_{S1}=150 stable structures including their corresponding natural frequencies as the inputs and nonlinear time history displacements of the top node as the outputs. To train and test the WRBFNN 100 and 50 samples are considered, respectively. The size of the WRBFNN is 3-100-500. In this example, the optimal values of a and b are obtained as 0.90 and 0.0, respectively. The results of testing the performance generality of the WRBFNN are shown in Figure 6. The time spent to train and test the WRBFNN is 3.0 min. The results imply that the generality of the WRBFNN is appropriate.

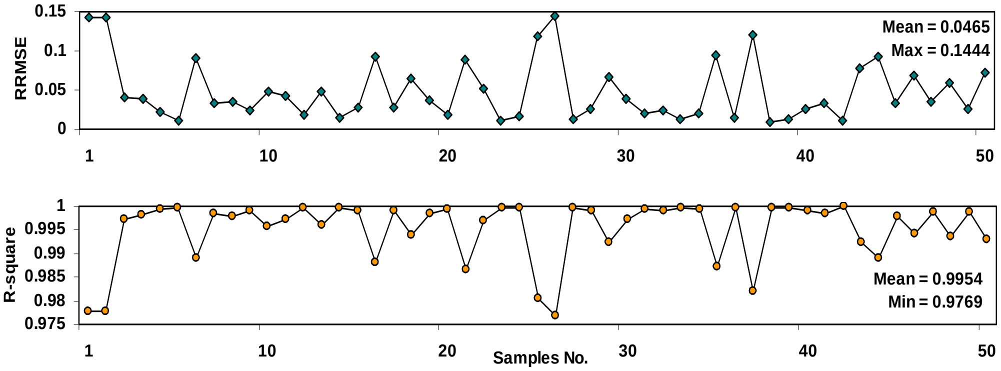

Figure 6. RRMSE and R^2 of the predicted responses by WRBFNN

6.5 Optimization Results

The optimization is carried out by the standard GA, with 30 individuals, and the PSO, with 30 particles, using the Exact Analysis (EA) and Approximate Analysis by HSM (HSM). The maximum number of generations is limited to 100. In each optimization case, ten independent optimal design processes (ten independent runs) are achieved and the best solution found, the average number of generations and the average time of optimization are considered as the final results. In the optimization process based on approximate analysis, to distinguish feasible and infeasible solutions, the criterion proposed by Vanderplaats [31] is involved: if the sum of the violated constraints is less than 0.005, the corresponding solution is feasible, otherwise the solution is infeasible. It is important to note that, in the optimization by GA and PSO using the HSM, the necessary responses during the optimization are predicted by the HSM. But, to assess the feasibility of the final optimal solution its responses are evaluated by the conventional FE time history analysis. The final results of optimization are given in Table 4.

Table 4. Optimum designs obtained by GA and PSO using EA and HSM

Element Groups No.	**GA**		**PSO**	
	EA	**HSM**	**EA**	**HSM**
1	1	1	1	1
2	1	1	1	1
3	2	4	2	1
4	3	2	2	2
5	7	5	2	2
6	2	3	2	2
7	9	9	2	4
8	2	2	2	2
9	1	1	1	1
Weight (kg)	1242.1	1237.3	1112.3	1083.8
The average number of generations	86	92	60	68
The sum of the violated constraints	0.0000	0.0000	0.0000	0.0050
The average time of Optimization (min)	3678.0	0.90	2566.0	0.60
Data generating time (min)	-	240.0	-	240.0
Training time (min)	-	5.50	-	5.50
Overall time (min)	3678.0	246.4	2566.0	246.1
RRMSE, R^2	-	0.0987, 0.9901	-	0.0923, 0.9912

It is demonstrated that the PSO is superior to the standard GA. It can be also observed that the best solution is obtained by PSO using HSM in terms of weight, time, and accuracy. The response of the best solution (PSO+HSM) is compared with its corresponding actual one in Figure 7.

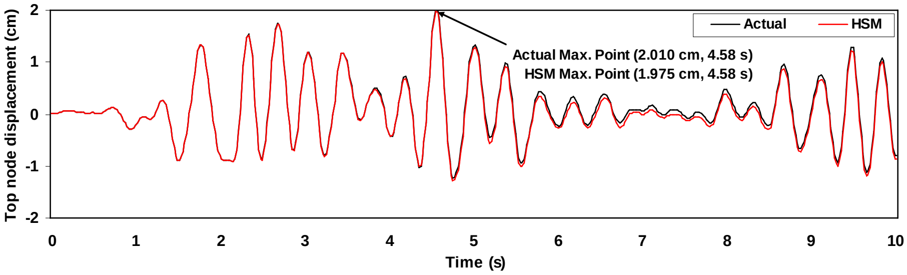

Figure 7. Comparison of best solution (PSO+HSM) response with its actual one

7. CONCLUDING REMARKS

In this paper a hybrid surrogate modeling based optimization algorithm is proposed to find optimum design of structures subjected to earthquake time history loading with nonlinear responses. Design optimization is performed by two well-known optimization algorithms: GA and PSO. In order to mitigate the computational rigors of the nonlinear time history analysis, an efficient hybrid surrogate model (HSA) is proposed to accurately predict the necessary nonlinear time history responses of the structures during the optimization process. In the HSM the main aspects of OLH, WT, RBF, and PNN are combined. The numerical results imply that the PSO possesses better computational performance comparing with the standard GA. Also it is observed that by employing PSO+HSM the overall time of optimization is 0.1 times of the time required by exact optimization (the PSO+EA procedure) while the errors due to all the approximations is small. Therefore, it can be finally concluded that the proposed methodology is a powerful tool to design optimization of structures subject to earthquake loading considering nonlinear behavior.

REFERENCES

1. Kennedy J. The particle swarm: social adaptation of knowledge, *Proceedings of the International Conference on Evolutionary Computation*, Piscataway, NJ: IEEE, 1977, pp. 303-308.
2. Holland JH. *Adaptation in Natural and Artificial Systems*, University of Michigan press, Ann Arbort, 1975.
3. Kaveh A, Kalatjari V. Topology optimization of trusses using genetic algorithm, force method and graph theory, *International Journal for Numerical Methods in Engineering*,

58(2003) 771-91.

4. Saka MP. Optimum design of pitched roof steel frames with haunched rafters by genetic algorith, *Computers and Structures*, **81**(2003) 1967-78.
5. Salajegheh E, Gholizadeh S. Optimum design of structures by an improved genetic algorithm using neural networks, *Advances in Engineering Software*, **36**(2005) 757-67.
6. Kaveh A, Talatahari S. A discrete particle swarm ant colony optimization for design of steel frames, *Asian Journal of Civil Engineering*, **9**(2008) 563-75.
7. Gholizadeh S, Salajegheh E. Optimal design of structures for time history loading by swarm intelligence and an advanced metamodel, *Computer Methods in Applied Mechanics and Engineering*, 198 (2009) 2936–49.
8. Kaveh A, Talatahari S. Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures, *Computers and Structures*, **87**(2009) 267-83.
9. Broomhead DS, Lowe D. Multi-variable functional interpolation and adaptive networks, *Complex Systems*, **2**(1988) 321–55.
10. Grossmann A, Morlet J. Decomposition of Hardi functions into square integrable wavelets of constant shape, *SIAM Journal of Mathematical Analysis*, **15**(1984)725–36.
11. Mckay MD, Beckman RJ, Conover WJ. A comparison of three methods for selecting values on input variables in the analysis of output from a computer code, *Technometrics*, **21**(1979) 439–45.
12. Gholizadeh S, Salajegheh J, Salajegheh E. An intelligent neural system for predicting structural response subject to earthquake, *Advances in Engineering Software*, **40**(2009) 630-9.
13. Gholizadeh S, Salajegheh E, Torkzadeh P. Structural optimization with frequency constraints by genetic algorithm using wavelet radial basis function neural network, *Journal of Sound and Vibration*, **312**(2008) 725–36.
14. Specht DF. Probabilistic neural networks, *Neural Networks*, **3**(1990) 109-18.
15. ANSYS Incorporated. ANSYS Release 10.0, 2006.
16. Lagaros ND, Fragiadakis M, Papadrakakis M, Tsompanakis Y. Structural optimization: A tool for evaluating dynamic design procedures, *Engineering Structures*, **28**(2006) 1623-33.
17. Arora JS, Optimization of structures subjected to dynamic loads, *Structural Dynamic Systems Computational Techniques and Optimization*, C.T. Leondes (Eds.), Gordon and Breach Science Publishers, 1999.
18. Kennedy J, Eberhart RC. Particle swarm optimization, *Proceedings of the IEEE International Conference on Neural Networks*, Perth, Australia, 1995, pp. 1942–1945.
19. Shi Y, Eberhart RC. A modified particle swarm optimizer, *Proceeding IEEE International Conference on Evolutionary Computation*, 1997, pp. 303–308.
20. Kathiravan R, Ganguli R. Strength design of composite beam using gradient and particle swarm optimization, *Composite Structures*, **81**(2007)471–9.
21. Fang KT, Li R, Sudjianto A. *Design and Modelling for Computer Experiments*, CRC Press, New York, 2006.
22. Audze P, Eglais V. New approach to planning out of experiments, *Problems of Dynamics and Strength*, **35**(1977) 104-7.

23. Ye KQ, Li W, Sudjianto A. Algorithmic construction of optimal symmetric Latin hypercubes, *Journal of Statistical Planning and Inference*, **90**(2000) 145-59.
24. Huang SL, Huang CS, Wen CM, Hsu YC. Nonparametric identification of a building structure from experimental data using wavelet neural networks, *Computer-Aided Civil and Infrastructure Engineering*, **18**(2003) 356-68.
25. Salajegheh E, Heidari A, Optimum design of structures against earthquake by wavelet neural network and filter banks, *Earthquake Engineering and Structural Dynamics*, **34**(2005) 67-82.
26. Martinet RK, Morlet J, Grossmann A. Analysis of sound patterns through wavelet transforms, *International Journal of Pattern Recognition and Artificial Intelligence*, **1**(1987) 273-302.
27. Wasserman PD. *Advanced Methods in Neural Computing*, Prentice Hall Company, Van Nostrand Reinhold, New York, 1993.
28. Salajegheh E, Gholizadeh S, Khatibinia M. Optimal design of structures for earthquake loads by a hybrid RBF-BPSO method, *Earthquake Engineering and Engineering Vibration*, **7**(2008) 13-24.
29. Seyedpoor SM, Salajegheh J, Salajegheh E, Gholizadeh S. Optimum shape design of arch dams for earthquake loading using fuzzy inference system and wavelet neural networks, *Engineering Optimization*, **41**(2009) 473-93.
30. Gholizadeh S, Salajegheh E. Optimal seismic design of steel structures by an efficient soft computing based algorithm, *Journal of Constructional Steel Research*, **66**(2010) 85-95.
31. Vanderplaats GN. *Numerical Optimization Techniques for Engineering Design*. 3rd ed. CO, USA: VR and D, Inc, 1999.