



OPTIMUM DESIGN OF STEEL STRUCTURES FOR EARTHQUAKE LOADING BY GREY WOLF ALGORITHM

S. Gholizadeh* and F. Fattahi

Department of Civil Engineering, Urmia University, Urmia, Iran

Received: 12 February 2015; **Accepted:** 5 April 2015

ABSTRACT

The main aim of this study is to propose an efficient computational strategy for optimization of steel structures subject to earthquake loading. To achieve the optimization task, two popular metaheuristics and the newly developed grey wolf algorithm (GWA) are employed. To reduce the computational burden of the process, radial basis function (RBF) and back propagation (BP) neural networks are used for evaluation the seismic responses of structures subject to three earthquakes. The numerical results show that GWA incorporating BP neural network provides the best results in comparison with the other ones.

Keywords: Optimization; metaheuristic; earthquake; time history analysis; steel structure; moment-resisting frame.

1. INTRODUCTION

Optimal design of structures for earthquake loading is one of the popular problems in the field of structural engineering. To evaluate the time history responses of structures, seismic design codes, such as UBC [1], specify that the structures should be analyzed for at least three horizontal ground motions. Therefore, the structural optimization for the earthquake loading is a computationally intensive task and it requires prohibitively high computing times for obtaining results from finite element analyses. Unreasonably high computing times could also prevent designers from comprehensively exploring the design space, and could ultimately result in unsuitable structures [2]. Therefore, utilizing an efficient computational strategy possessing the global search ability and reasonable computational effort is vital for the design optimization process of the structures subject to earthquake loading. One of the main aims of this study is to reduce the computational effort of the optimal design process of steel moment resisting frames subject to earthquake loading by an efficient soft computing based strategy.

During the last decade, structural optimization problems have been solved using gradient-

*E-mail address of the corresponding author: s.gholizadeh@urmia.ac.ir (S. Gholizadeh)

based algorithms. As the mathematical programming based methods need gradient calculations, a considerable part of the optimization process is devoted to the sensitivity analysis and the computational work of these methods is usually high. Employing stochastic search techniques allows exploration of a larger fraction of the design space in comparison with gradient-based optimization methods. In order to improve the global search ability and to achieve the optimization task in a more easy and flexible way, a variety of metaheuristic optimization methods inspired by nature were developed. The metaheuristics demonstrate their efficiency in many of the structural optimization problems and this is why these methods have been extensively employed in the field of structural engineering. During the last years, genetic algorithm (GA) [3] and harmony search algorithm (HSA) [4] have been widely employed for solving many optimization problems in the field of civil and structural engineering. Recently, Mirjalili et. al. [5] proposed grey wolf algorithm (GWA) as a new metaheuristic. As the structural optimization for earthquake loading is an interesting and challenging problem area in structural engineering [6-12] in this study the computational performance of GWA is compared with that of GA and HSA for optimization of steel structures subject to earthquake loading.

In order to reduce the computational time of the structural optimization for the earthquake loading, neural network techniques are the best candidate. They are particularly suitable for problems too complex to be modeled and solved by classical mathematics and traditional procedures. In recent years, neural networks have been widely used to solve complex problems in the fields of civil and structural engineering. In this work, back propagation (BP) and radial basis function (RBF) neural networks are trained to predict the demand-capacity ratio (DCR) of structural elements and the maximum inter-story drift of structures subject to three earthquake records during the optimization process.

Two steel frame structures are optimized by the mentioned metaheuristics incorporating neural networks to predict the required structural responses during the optimization process. The numerical results indicate that the GWA possesses the best computational performance compared with that of the mentioned metaheuristics.

2. PROBLEM FORMULATION

For optimal design of a steel frame including ne members collected in ng design groups, the design variables of each design group are usually selected from a given standard profile list. In this case, the optimization problem may be formulated as follows:

$$\text{Minimize: } w(X) = \sum_{i=1}^{ng} \rho_i A_i \sum_{j=1}^{nm} L_j \quad (1)$$

$$\text{Subject to: } g_k(X) \leq 0, \quad k = 1, 2, \dots, nc \quad (2)$$

$$X = \{x_1 \quad x_2 \quad \dots \quad x_i \quad \dots \quad x_{ng}\}^T \quad (3)$$

where x_i is an integer value expressing the sequence numbers of steel sections assigned to i th group; w represents the weight of the frame, ρ_i and A_i are weight of unit volume and cross-sectional area of the i th group section, respectively; nm is the number of elements

collected in the i th group; L_j is the length of the j th element in the i th group; $g_k(X)$ is the k th behavioral constraint.

In such optimization problem, lateral inter-story drift constraint is usually taken as:

$$g_l = \frac{\theta_l}{R_l} - 1 \leq 0, \quad l = 1, \dots, ns \quad (4)$$

where θ_l is the inter-story drift; R_l is the inter-story drift index permitted by the code of practice and ns is the total number of stories.

The DCR constraints of structural elements subjected to axial and flexural stresses are computed as follows [13]:

$$DCR = \begin{cases} \left[\frac{P_u}{2\phi_c P_n} + \left(\frac{M_{ux}}{\phi_b M_{nx}} + \frac{M_{uy}}{\phi_b M_{ny}} \right) \right] & \text{if } \frac{P_u}{\phi_c P_n} < 0.2 \\ \left[\frac{P_u}{\phi_c P_n} + \frac{8}{9} \left(\frac{M_{ux}}{\phi_b M_{nx}} + \frac{M_{uy}}{\phi_b M_{ny}} \right) \right] & \text{if } \frac{P_u}{\phi_c P_n} \geq 0.2 \end{cases} \quad (5)$$

$$g_{DCR} = DCR_m - 1 \leq 0, \quad m = 1, \dots, ne \quad (6)$$

where P_u is the required strength; P_n is the nominal axial strength (tension or compression); ϕ_c is the resistance factor; M_{ux} and M_{uy} are the required flexural strengths in the x and y directions; respectively; M_{nx} and M_{ny} are the nominal flexural strengths in the x and y directions; and $\phi_b = 0.9$ is the flexural resistance reduction factor.

The effective length factor, K , for beam and bracing members is taken equal to unity. This parameter for columns is calculated from the approximate Eqs. (7) for unbraced frames, which are accurate to within about -1.0% and +2.0% of the exact results [14]:

$$K = \sqrt{\frac{1.6G_A G_B + 4(G_A + G_B) + 7.5}{G_A + G_B + 7.5}} \quad (7)$$

where G_A and G_B refer to stiffness ratio or relative stiffness of a column at its two ends.

In order to satisfy practical demands, geometric constraints should be considered in beam-column framing joints as follows for 2D frames:

$$g_B = \left(\frac{b_{fb}}{b_{fc}} \right)_n - 1 \leq 0, \quad n = 1, \dots, nj \quad (8)$$

where b_{fb} and b_{fc} are the flange width of beam and column, respectively; nj is the number of joints.

In the present work, the exterior penalty function method (EPFM) [15] is employed to handle the design constraints. The EPFM transforms the basic constrained optimization

problem into the unconstrained formulation. In this case, the pseudo unconstrained objective function can be represented as follows:

$$\Psi(X, r) = w(X) \left(1 + r \sum_{l=1}^{ns} (\max\{0, g_d\})^2 + r \sum_{m=1}^{ne} (\max\{0, g_{DCR}\})^2 + r \sum_{n=1}^{nj} (\max\{0, g_B\})^2 \right) \quad (9)$$

where Ψ and r are the pseudo objective function and a penalty parameter, respectively.

For minimizing the above mentioned pseudo objective function a number of popular metaheuristic algorithms are employed in the present study. The theoretical background of these metaheuristics is explained in the next section.

3. METAHEURISTICS

The main idea behind designing the metaheuristic algorithms is to tackle complex optimization problems where other optimization methods have failed to be effective [16]. Metaheuristics are applied to a very wide range of problems and they mimic natural metaphors to solve complex optimization problems [17]. In this study, GA and HSA as popular metaheuristics and GWA as a new metaheuristic are applied to find optimum design of steel frames subject to earthquake loading.

3.1 Genetic algorithm

GA tries to simulate biological evolution. It searches by simulating evolution, starting from an initial set of solutions and generating successive generations of solutions. A simple GA proceeds by randomly generating an initial population. The next generation is evolved from this population by performing reproduction, crossover, and mutation operations. Reproduction operator reproduces the next generation based on the statistics of current population. In this way, the weak designs are removed and the strong ones are transformed to the next generation. In the crossover operation, two members of the population are randomly selected, as parents, and two new offsprings are produced by exchanging a part of parents' string at a randomly selected position with a specified probability of crossover. Finally, with a probability of mutation, certain digits of the chromosomes are altered. In this way, the population takes its final form in the current generation. After several generations, the best individual of the population is considered as the final solution of the algorithm.

The stochastic nature of the method and using a population of design points in each generation usually give rise to the global optimum. However, if exploration and exploitation abilities are not properly balanced in the GA, a global optimum may not be guaranteed, although, near-optimal solutions are found easily. The stochastic nature of standard GA makes the convergence of the procedure slow. In general, the standard GA is not convenient to find the solution of large scaled and complex problems. Up to now GA and its improved versions have been extensively employed by researchers to efficiently tackle the complex problems in the area of structural engineering.

3.2 Harmony search algorithm

HSA is based on the musical performance process that achieves when a musician searches for a

better state of harmony. In the process of musical production a musician selects and brings together number of different notes from the whole notes and then plays these with a musical instrument to find out whether it gives a pleasing harmony. The musician then tunes some of these notes to achieve a better harmony.

For implementation of HS, at first a harmony memory (HM), the harmony considering rate (HMCR), the pitch adjusting rate (PAR) and the maximum number of searches should be specified. To improvise new HM, a new harmony vector is generated. Thus the new value of the i th design variable can be chosen from the possible range of i th column of the HM with the probability of HMCR or from the entire possible range of values with the probability of $1 - \text{HMCR}$ as follows:

$$x_i^{\text{new}} = \begin{cases} x_i^j \in \{x_i^1, x_i^2, \dots, x_i^{\text{HMS}}\}^T & \text{with the probability of HMCR} \\ x_i \in \Delta_i & \text{with the probability of } (1 - \text{HMCR}) \end{cases} \quad (10)$$

where Δ_i is the set of the potential range of values for i th design variable.

Pitch adjusting is performed only after a value has been chosen from the HM as follows:

$$\text{pitch adjustment of } x_i^{\text{new}} ? \begin{cases} \text{Yes} & \text{with the probability of PAR} \\ \text{No} & \text{with the probability of } (1 - \text{PAR}) \end{cases} \quad (11)$$

If the pitch-adjustment decision for x_i^{new} is "Yes", then a neighbouring value with the probability of $\text{PAR} \times \text{HMCR}$ is taken for it as follows:

$$x_i^{\text{new}} \leftarrow \begin{cases} x_i^{\text{new}} \pm u(-1, +1) \times bw & \text{with the probability of } \text{PAR} \times \text{HMCR} \\ x_i^{\text{new}} & \text{with the probability of } \text{PAR} \times (1 - \text{HMCR}) \end{cases} \quad (12)$$

where $u(-1, +1)$ is a uniform distribution between -1 and $+1$; also bw is an arbitrary distance bandwidth for the continuous design variables.

If x_i^{new} is better than the worst vector in the HM, the existing worst harmony is replaced by the new harmony.

3.3 Grey wolf algorithm

GWA is a new metaheuristic and has been proposed by Mirjalili *et. al.* [5] based on the leadership hierarchy and hunting mechanism of grey wolves in nature. In the process of GWA, the leadership hierarchy and the hunting process are simulated.

In the wolf's pack the leaders are usually a male and a female, called alpha and their decisions are dictated to the pack. The second level in the hierarchy of grey wolves is beta. The betas are subordinate wolves that help the alpha in decision-making or other pack activities. The beta wolf is probably the best candidate to be the alpha in case one of the alpha wolves passes away or becomes very old. The lowest ranking grey wolf is omega. The omega plays the role of scapegoat. Omega wolves always have to submit to all the other dominant wolves. They are the last wolves that are allowed to eat. If a wolf is not an alpha, beta, or omega, he/she is

called subordinate (or delta in some references). Delta wolves have to submit to alphas and betas, but they dominate the omega [5].

Hunting as another social behavior of grey wolfs comprises three main phases. The first phase includes tracking, chasing, and approaching the prey. In the second one the grey wolfs pursue, encircle, and harass the prey until it stops moving and in the last phase they attack towards the prey [5]. In [5] the mentioned hunting technique and the social hierarchy of grey wolfs have been mathematically modeled to propose GWA.

In designing GWA, the first, second and third best solutions are considered as α , β and δ wolves, respectively while the rest of the candidate solutions are considered as ω . In the framework of GWA, ω wolves follow α , β and δ wolves during the optimization process.

The following equations are used to model the encircling behavior of grey wolfs [5]:

$$A_i = 2\bar{A}_i \cdot R_{1i} - \bar{A}_i \quad (13)$$

$$C_i = 2R_{2i} \quad (14)$$

$$D_i = |C_i \cdot X_p^t - X_i^t| \quad (15)$$

$$X_i^{t+1} = X_p^t - A_i \cdot D_i \quad (16)$$

where R_{1i} and R_{2i} are random vectors in $[0,1]$; \bar{A}_i is a vector that its components are linearly decreased from 2 to 0 during the optimization process; A_i and C_i are coefficient vectors; X_p^t is the prey in iteration t ; X_i^t is the i th grey wolf in iteration t .

For simulation of the hunting behavior of grey wolfs, it is supposed that the alpha, beta, and delta have better knowledge about the potential location of prey. Therefore, the first three best solutions obtained so far should be saved and the other wolves in the pack update their positions according to the position of the best ones (around the prey) as follows [5]:

$$D_\alpha = |C_1 \cdot X_\alpha^t - X^t| \quad (17)$$

$$D_\beta = |C_2 \cdot X_\beta^t - X^t| \quad (18)$$

$$D_\delta = |C_3 \cdot X_\delta^t - X^t| \quad (19)$$

$$X_1^t = X_\alpha^t - A_1 \cdot D_\alpha \quad (20)$$

$$X_2^t = X_\beta^t - A_2 \cdot D_\beta \quad (21)$$

$$X_3^t = X_\delta^t - A_3 \cdot D_\delta \quad (22)$$

$$X^{t+1} = \frac{1}{3} \sum_{j=1}^3 X_j^t \quad (23)$$

The final step in hunting process of grey wolfs is attacking prey as soon as it stops moving. In GWA, decreasing the values of \bar{A}_i components from 2 to 0 during the optimization process simulates approaching the prey and provides the exploration ability of the algorithm. Also, the exploitation ability of the GWA comes from the random components of the C vector.

4. NEURAL NETWORKS

Neural networks are robust computational tools that have the ability to extract the underlying complex dependencies from measured data. The most important point about a properly trained neural network is that it requires a trivial computational effort to produce an approximate solution. Such approximations appear to be valuable in situations where the actual response computations are intensive in terms of computing time and a quick estimation is required. In the present work, radial basis function (RBF) and back propagation (BP) neural networks are employed to predict the responses of structures subjected to earthquake time history loading.

4.1 Radial basis function

Radial basis function (RBF) neural networks are two layers feed forward networks. The hidden layer consists of RBF neurons with Gaussian activation functions. The outputs of RBF neurons have significant responses to the inputs only over a range of values of Y called the receptive field. The size of the receptive field is determined by the value of σ . Activation function of RBF neurons is as follows:

$$f^{\text{RBF}}(y) = \exp\left(-0.5\left(\frac{y}{\sigma}\right)^2\right) \quad (24)$$

where y is an input, and σ is radius of receptive field of RBF neuron.

The value of σ allows the sensitivity of the RBF neurons to be adjusted. During the training, the σ value of RBF neurons is such determined as the neurons could cover the input space properly. To train the hidden layer of RBF networks no training is accomplished and the transpose of training input matrix is taken as the layer weight matrix [18].

$$W_1^{\text{RBF}} = Y^T \quad (25)$$

where, W_1^{RBF} and Y^T are input layer weight and training input matrices, respectively.

In order to adjust output layer weights, a supervised training algorithm is employed. The output layer weight matrix is calculated from the following equation:

$$W_2^{\text{RBF}} = \Delta^{-1}T \quad (26)$$

in which T is the target matrix, Δ is the outputs of the hidden layer and W_2^{RBF} is the output layer weight matrix.

In [12] RBF neural network has been used to predict the structural time history responses for earthquake loading.

4.2 Back propagation

Training of multi-layer perceptron, usually called back propagation (BP) neural network, is achieved by employing traditional gradient-based methods. The basic equation in training phase of BP is as follows:

$$W_{k+1}^{\text{BP}} = W_k^{\text{BP}} - \alpha_k G_k \quad (27)$$

where W_k^{BP} is current weights, G_k is the current gradient, and α_k is the learning rate.

Activation function of BP neural networks is hyperbolic tangent sigmoid function and it is given as follows:

$$f^{\text{BP}}(y) = \frac{e^y - e^{-y}}{e^y + e^{-y}} \quad (28)$$

Levenberg–Marquardt (LM) [19] algorithm, which is usually applied to train BP neural network, uses the approximate Hessian matrix in the following Newton-like update:

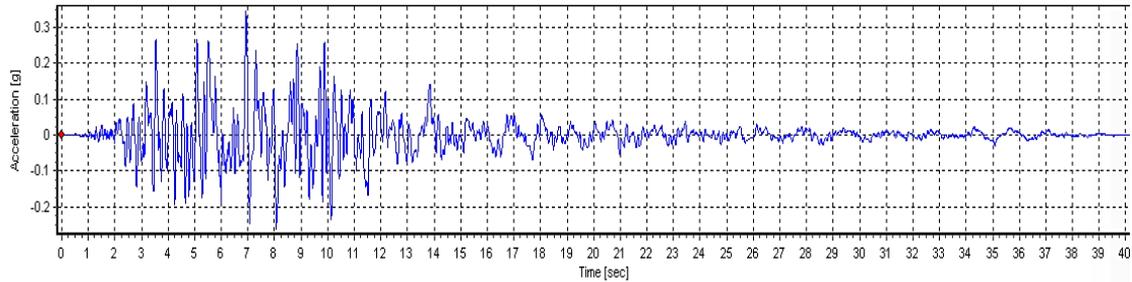
$$W_{k+1}^{\text{BP}} = W_k^{\text{BP}} - [J^T J + \mu I]^{-1} J^T E \quad (29)$$

where J is the Jacobian matrix that contains first derivatives of the network errors with respect to the weights; E is a vector of network errors; and μ is a correction factor.

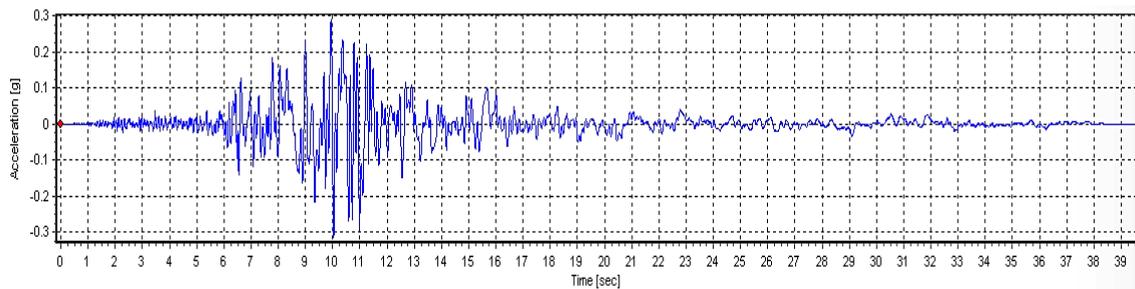
BP neural network has been employed to predict the structural time history responses for earthquake loading in [20].

5. METHODOLOGY

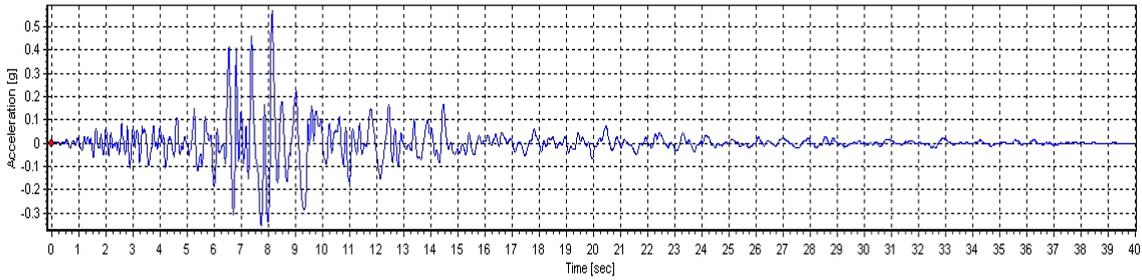
In this study, optimum design of steel frames is achieved according to Iranian Code of Practice for Seismic Resistant Design of Buildings (ICPSRDB) [21] by neural network incorporated metaheuristics for earthquake loading. The ground motion records of three earthquakes of Kobe, Imperial Valley and Northridge, shown in Fig. 1, are selected.



(a)



(b)



(c)

Figure 1. Earthquake records of (a) Kobeh, (b) Imperial Valley and (c) Northridge

These records are then scaled based on the design spectrum of [21]. Here the elastic linear time history analysis is conducted to determine the seismic responses of the structures.

Cross-sectional area assignments of the structural members are selected from the standard IPE profiles available in Table 1.

Table 1: The available standard IPE profiles

| No. | Profile | No. | Profile |
|-----|---------|-----|---------|
| 1 | IPE 120 | 9 | IPE 300 |
| 2 | IPE 140 | 10 | IPE 330 |
| 3 | IPE 160 | 11 | IPE 360 |
| 4 | IPE 180 | 12 | IPE 400 |
| 5 | IPE 200 | 13 | IPE 450 |
| 6 | IPE 220 | 14 | IPE 500 |
| 7 | IPE 240 | 15 | IPE 550 |
| 8 | IPE 270 | 16 | IPE 600 |

Neural network training is the first phase of the computational strategy of this study. To train RBF and BP neural networks, cross-sectional area assignments of a number of steel structures, satisfying the geometric constraints, are randomly selected. The selected structures are analyzed for the scaled three earthquake records by using OpenSees [22]. Then, their maximum inter-story drift and maximum DCR of each element groups are evaluated. Considering the sequence numbers of steel sections assigned to element groups of the generated steel structures (X vector defined by Eq. (3)) as the input vector and their corresponding maximum inter-story drift and maximum DCR of each element groups as the components of the output vector, RBF and BP neural networks are trained using MATLAB [23]. The absolute percentage error (APE) between the actual response (λ^{act}) and the approximate one (λ^{app}) is calculated as follows:

$$APE = 100 \left| \frac{\lambda^{\text{act}} - \lambda^{\text{app}}}{\lambda^{\text{act}}} \right| \quad (30)$$

To evaluate the prediction accuracy of the neural networks in testing mode, mean absolute percentage error (*MAPE*) and standard deviation (*SD*) of errors are computed as follows:

$$MAPE = \frac{1}{n_{ts}} \sum_{i=1}^{n_{ts}} APE_i \quad (31)$$

$$SD = \sqrt{\frac{1}{n_{ts}} \sum_{i=1}^{n_{ts}} (APE_i - MAPE)^2} \quad (32)$$

where n_{ts} is the number of testing samples.

The second phase of the computational strategy of this study is optimization of the steel frames employing metaheuristics in which the time history analysis box is replaced by the properly trained neural networks and this results in substantially decrease of the computational burden of the optimization process.

6. NUMERICA RESULTS

In order to compare the computational performance of the mentioned metaheuristics for optimization of steel structures subject to earthquake loading, two numerical examples are presented. For the first and second examples a database, respectively, including 200 and 400 samples is generated and from which 90% and 10% of samples are used for training and testing, respectively. Also, for the employed metaheuristics, population size and maximum number of iterations are 30 and 100, respectively. If the fundamental period of the structure is less than 0.7 sec. the permitted inter-story drift index is 2.5%, otherwise it is 2% [21]. For the numerical examples, Young's modulus, weight density and yield stress are 2.1×10^{10} kg/m², 7850 kg/m³ and 3.515×10^7 kg/m², respectively.

6.1 First example: 6-story 1-bay planar steel frame

The 6-story 1-bay steel frame shown in Fig. 2 is the first example of this study. As shown in Fig. 2, there are 6 design variables in this example including 3 column sections and 3 beam sections related variables. The applied dead and live loads on the beams are 3600 kg/m and 1200 kg/m, respectively.

Since in this example there are six design variables, the input vector of the neural network models includes 6 components. Moreover, the maximum drift and maximum DCRs of six element groups are the components of the output vector. As for training RBF, with one hidden layer, 180 samples are used, its architecture is 6-180-7. For training BP two hidden layers are employed and the best results are obtained by using 10 and 15 neurons respectively in the first and second hidden layers, thus its architecture is 6-10-15-7.

The performance generality of the neural networks in testing mode are compared in Table 2 in terms of *MAPE* and *SD*. Average *MAPE* and *SD* for RBF are 3.39% and 3.85%, respectively while for BP are 3.01% and 2.32%, respectively. Furthermore, in training mode, average *MAPE* and *SD* for RBF are 0.47% and 0.45%, respectively while for BP are 0.42% and 0.28%, respectively. These results demonstrate the superiority of the BP over the RBF.

In the present example, the time spent to train the neural networks including data generation

time is equal to 53.0 min.

As the results demonstrate that BP possesses better computational performance in comparison with RBF, the BP neural network is incorporated in the optimization process to evaluate the required seismic responses.

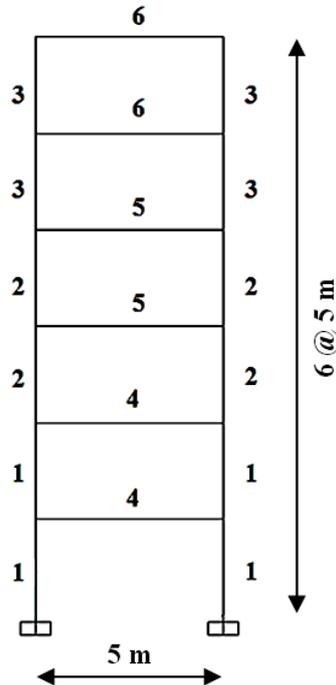


Figure 2. The 6-story steel frame

Table 2: Comparison the testing results of RBF and BP for the 6-story steel frame

| Seismic Responses | RBF | | BP | |
|-------------------|----------|--------|----------|--------|
| | MAPE (%) | SD (%) | MAPE (%) | SD (%) |
| θ_{\max} | 3.92 | 2.94 | 6.11 | 4.54 |
| DCR_1 | 3.57 | 4.25 | 2.26 | 1.48 |
| DCR_2 | 3.01 | 3.05 | 2.32 | 1.74 |
| DCR_3 | 3.64 | 4.62 | 3.25 | 2.89 |
| DCR_4 | 3.63 | 4.30 | 2.51 | 2.01 |
| DCR_5 | 3.54 | 4.37 | 1.71 | 1.44 |
| DCR_6 | 2.41 | 3.42 | 2.86 | 2.15 |
| Average (%) | 3.39 | 3.85 | 3.01 | 2.32 |

Optimization process is implemented by GA, HSA and GWA incorporating the trained BP neural network and the results are compared in Table 3 with the engineering design achieved by SAP2000 [24] software.

In addition, Fig. 3 depicts the optimization convergence histories of GA, HSA and GWA metaheuristics.

Table 3: Engineering and optimum designs of the 6-story steel frame

| Design Variable No. | Engineering Design | Optimum Design | | |
|---------------------|--------------------|----------------|---------|---------|
| | | GA | HSA | GWA |
| 1 | IPE 450 | IPE 450 | IPE 450 | IPE 400 |
| 2 | IPE 360 | IPE 360 | IPE 360 | IPE 360 |
| 3 | IPE 330 | IPE 330 | IPE 300 | IPE 300 |
| 4 | IPE 360 | IPE 360 | IPE 360 | IPE 360 |
| 5 | IPE 360 | IPE 360 | IPE 360 | IPE 360 |
| 6 | IPE 330 | IPE 300 | IPE 300 | IPE 300 |
| Weight (kg) | 3838.02 | 3768.94 | 3686.05 | 3551.34 |
| Time (sec.) | 30.0 | 36.9 | 34.8 | 32.5 |

It can be observed from the presented results that among the employed metaheuristics, GWA is the best one. Furthermore, the weight of the optimum design found by GWA is 7.47% lighter than the weight of the engineering design.

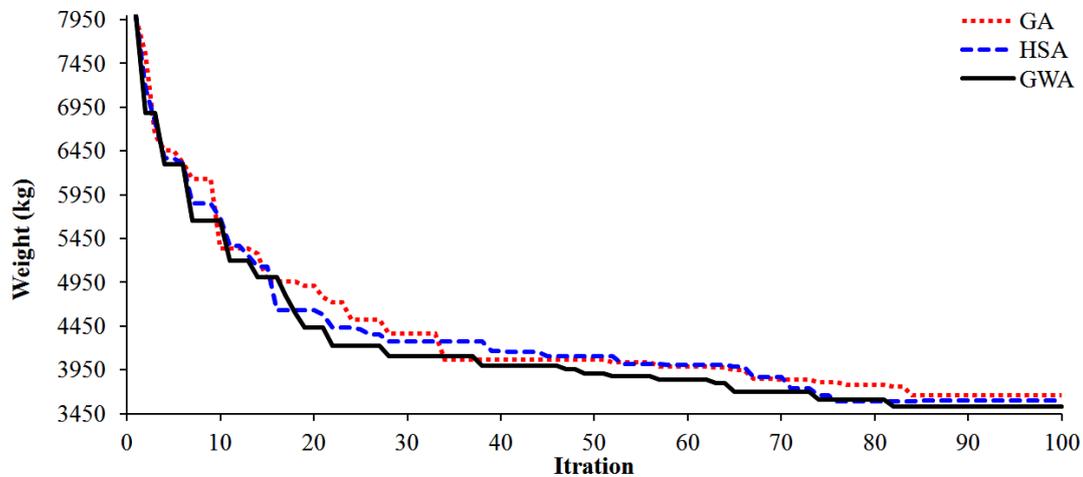


Figure 3. Convergence histories of metaheuristics for optimization of 6-story frame

The optimum solution found by GWA incorporated BP is analyzed for the three earthquake records and the results are as follows: $\theta_{\max}^{\text{act}} = 2.39\%$ and $DCR_{\max}^{\text{act}} = 0.75$. In comparison with $\theta_{\max}^{\text{app}} = 2.46\%$ and $DCR_{\max}^{\text{app}} = 0.78$ it can be concluded that the solution is feasible and the trained BP neural network provides proper approximation accuracy in the design space especially in the region containing the optimum solution. Furthermore, inter-story drifts (θ) and maximum DCR of each element group for engineering design and optimal design found by GWA are compared in Fig. 4.

It should be noted that, as the time spent for each time history analysis by OpenSees is 5.0 sec., without employing neural networks to evaluate the necessary structural responses, the time spent to optimum design is about 750 min. and this emphasizes on the key role of neural networks for considerably shortening the optimization process of structures subjected to earthquake loading.

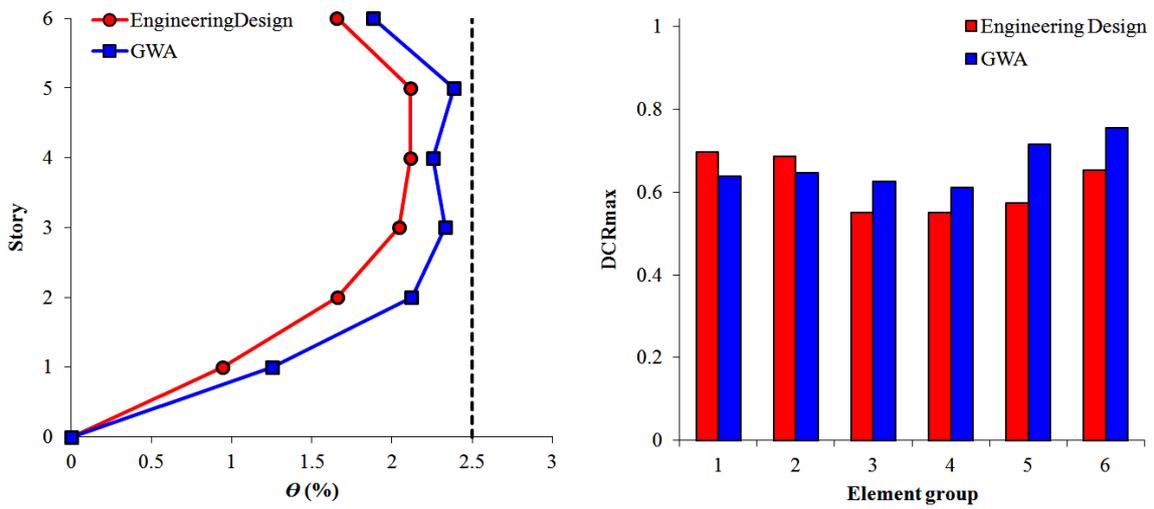


Figure 4. θ and DCR_{max} for engineering design and GWA optimal design of 6-story frame

6.2 Second example: 12-story 3-bay planar steel frame

Fig. 5 depicts the geometry and element grouping details of 12-story 3-bay steel frame.

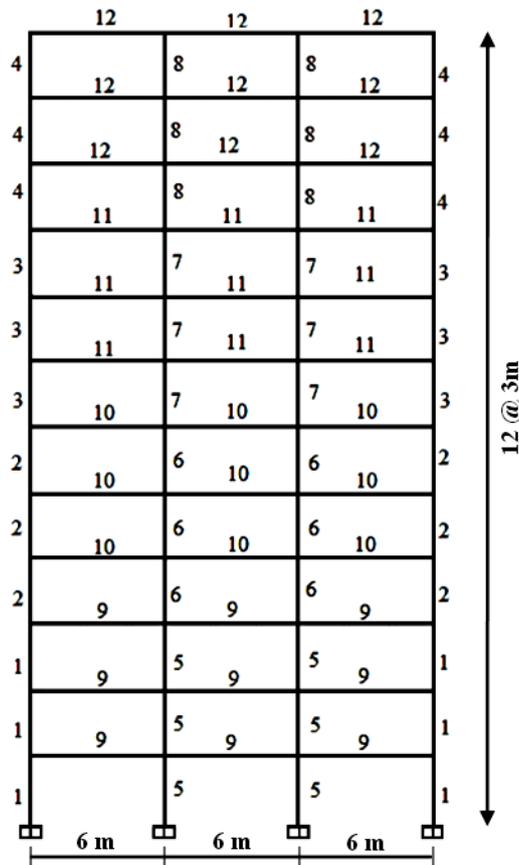


Figure 5. The 12-story steel frame

There are 12 design variables in this example including 8 column sections and 4 beam sections related variables. The applied dead and live loads on the beams are 3000 kg/m and 1000 kg/m, respectively.

Table 4 reports the *MAPE* and *SD* for the trained RBF and BP neural networks in testing mode. The architecture of the RBF and BP neural networks are 12-360-13 and 12-15-15-13, respectively. In the present example, the time of training including data generation is about 155.0 min.

Table 4: Comparison the testing results of RBF and BP for the 12-story steel frame

| Seismic Responses | RBF | | BP | |
|-------------------|-----------------|---------------|-----------------|---------------|
| | <i>MAPE</i> (%) | <i>SD</i> (%) | <i>MAPE</i> (%) | <i>SD</i> (%) |
| θ_{\max} | 4.28 | 3.46 | 6.22 | 5.87 |
| DCR_1 | 3.35 | 2.67 | 1.69 | 1.45 |
| DCR_2 | 2.27 | 2.35 | 2.27 | 1.98 |
| DCR_3 | 2.58 | 2.49 | 2.31 | 2.50 |
| DCR_4 | 3.93 | 2.97 | 1.88 | 1.65 |
| DCR_5 | 2.2 | 2.30 | 1.59 | 1.16 |
| DCR_6 | 3.42 | 2.41 | 2.32 | 2.26 |
| DCR_7 | 7.20 | 6.45 | 2.62 | 2.36 |
| DCR_8 | 6.66 | 5.12 | 2.93 | 2.86 |
| DCR_9 | 2.98 | 3.67 | 1.86 | 1.44 |
| DCR_{10} | 2.34 | 2.11 | 1.87 | 1.42 |
| DCR_{11} | 3.61 | 3.89 | 1.73 | 1.65 |
| DCR_{12} | 2.46 | 2.62 | 1.02 | 1.01 |
| Average (%) | 3.64 | 3.27 | 2.33 | 2.14 |

Table 5: Engineering and optimum designs of the 12-story steel frame

| Design Variable No. | Engineering Design | Optimum Design | | |
|---------------------|--------------------|----------------|----------|----------|
| | | GA | HSA | GWA |
| 1 | IPE 500 | IPE450 | IPE400 | IPE450 |
| 2 | IPE 400 | IPE400 | IPE360 | IPE360 |
| 3 | IPE 360 | IPE360 | IPE360 | IPE330 |
| 4 | IPE 300 | IPE300 | IPE300 | IPE300 |
| 5 | IPE 550 | IPE600 | IPE600 | IPE600 |
| 6 | IPE 550 | IPE500 | IPE500 | IPE500 |
| 7 | IPE 450 | IPE450 | IPE450 | IPE400 |
| 8 | IPE 330 | IPE330 | IPE300 | IPE300 |
| 9 | IPE 300 | IPE300 | IPE300 | IPE270 |
| 10 | IPE 300 | IPE300 | IPE300 | IPE270 |
| 11 | IPE 330 | IPE300 | IPE300 | IPE270 |
| 12 | IPE 300 | IPE300 | IPE300 | IPE300 |
| Weight (kg) | 20390.53 | 19623.74 | 19130.61 | 17983.25 |
| Time (sec.) | 56.0 | 68.0 | 73.0 | 59.0 |

The results of Table 4 indicate that the approximation accuracy of the BP is better than that of the RBF. Therefore, during the optimization process the BP is utilized to predict the required structural responses.

Table 5 compares the optimum designs found by GA, HSA and GWA with the engineering design. Convergence histories of GA, HSA and GWA metaheuristics are shown in Fig. 6.

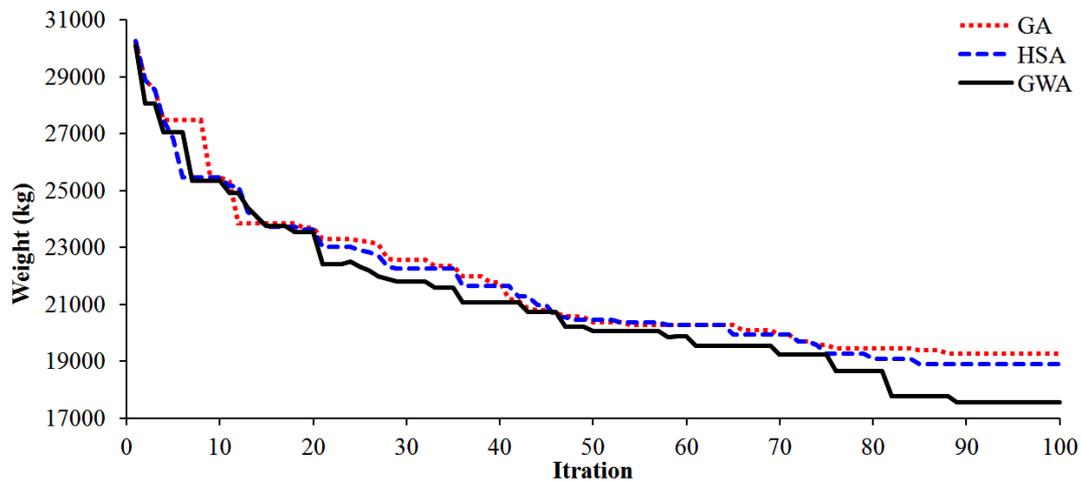


Figure 6. Convergence histories of metaheuristics for optimization of 12-story frame

It is evident from the results that GWA is the best metaheuristic among the all in terms of optimal weight and convergence rate. Moreover, GWA converges to an optimal solution which is 11.80% lighter than the engineering design. This optimum solution is analyzed for the three records and it is observed that $\theta_{\max}^{\text{act}} = 1.97\%$ and $DCR_{\max}^{\text{act}} = 0.99$. As $\theta_{\max}^{\text{app}} = 1.94\%$ and $DCR_{\max}^{\text{app}} = 0.99$, the feasibility of the solution is proven and this is due to the appropriate performance generality of the trained BP neural network in the design space.

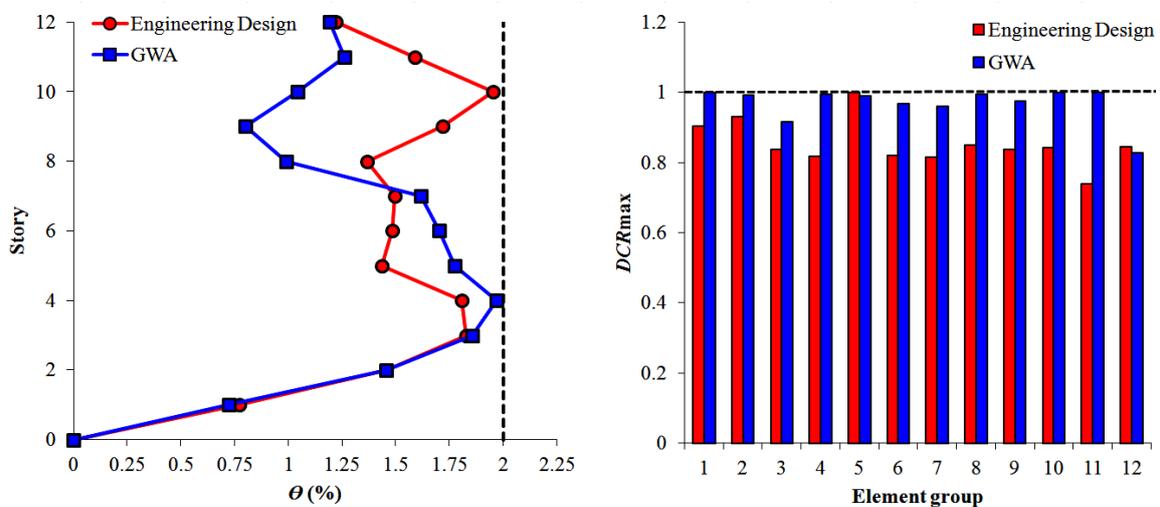


Figure 7. θ and DCR_{\max} for engineering design and GWA optimal design of 12-story frame

Fig. 7 compares inter-story drifts (θ) and maximum DCR of element groups for engineering and GWA optimal designs.

The time of each time history analysis of the 12-story steel frame by OpenSees is 7.5 sec. and therefore the time spent to optimum design of this structure without employing neural networks is about 1130 min. In this case, the overall time of optimization by using neural network (including data generation time) is about 86% reduced.

7. CONCLUSIONS

Computational performance of the newly developed GWA is investigated for optimization of steel frame structures subject to earthquake time history loading in comparison with popular metaheuristics such as GA and HSA. As the computational cost of structural optimization for earthquake time history loading is very expensive, neural network techniques are employed to reduce the computational burden. RBF and BP neural networks are trained to predict the structural seismic responses and due to better accuracy of the BP neural network it is utilized during the optimization process to evaluate the necessary responses. Two numerical examples, including a 6-story 1-bay planar steel frame and a 12-story 3-bay planar steel frame are presented. The numerical results demonstrate the efficiency of the GWA compared to the other metaheuristics in terms of found optimal structural weight and convergence rate. In the case of first example, the optimal weight found by GWA is 5.77% and 3.65% less than those of the GA and HSA, respectively. In the case of second one, GWA converges to an optimal solution which is 8.36% and 6.00% lighter than the solutions found by GA and HSA, respectively. Moreover, comparison of optimal designs of GWA with engineering designs of two presented examples reveal that the optimal designs are respectively 7.47% and 11.80% lighter than the engineering designs. It can be finally concluded that the GWA is an efficient metaheuristic for optimal design of steel structures subject to earthquake time history loading.

REFERENCES

1. International conference of building officials, *Uniform Building Code*, 2(1997).
2. 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.
3. Holland JH. *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, 1975.
4. Geem ZW, Kim JH, Loganathan GV. A new heuristic optimization algorithm: harmony search, *Simulations*, 76(2001) 60-8.
5. Mirjalili SA, Mirjalili SM, Lewis A. Grey wolf optimizer, *Advances in Engineering Software*, 69(2014) 46-61.
6. Kaveh A, Azar BF, Hadidi A, Sorooshi FR, Talatahari S. Performance-based seismic design of steel frames using ant colony optimization, *Journal of Constructional Steel Research*, 66(2010) 566-74.

7. Kaveh A, Laknejadi K, Alinejad B. Performance-based multi-objective optimization of large steel structures, *Acta Mechanica*, **223**(2012) 355-69.
8. Kaveh A, Nasrollahi A. Performance-based seismic design of steel frames utilizing charged system search optimization, *Applied Soft Computing*, **22C**(2014) 213-21.
9. Kaveh A, Kalateh-Ahani M, Fahimi-Farzam M. Life-cycle cost optimization of steel moment-frame structures: performance-based seismic design approach, *Earthquakes and Structures*, **7**(2014) 271-94.
10. Kaveh A, Kalateh-Ahani M, Fahimi-Farzam M. Damage-based optimization of large-scale steel structures, *Earthquakes and Structures*, **7**(2014) 1119-39.
11. Gholizadeh S. Performance-based optimum seismic design of steel structures by a modified firefly algorithm and a new neural network, *Advances in Engineering Software*, **81**(2015) 50-65.
12. Gholizadeh S, Salajegheh E. Optimal design of structures for earthquake loading by self organizing radial basis function neural networks, *Advances in Structural Engineering*, **13**(2010) 339-56.
13. American Institute of Steel Construction (AISC). *Manual of Steel Construction-Load Resistance Factor Design*, (3 rd Edition), AISC, Chicago, IL, USA, 1991.
14. Hellesland J. *Review and Evaluation of Effective Length Formulas*, Research Report, no. 94–102, University of Oslo, 1994.
15. Vanderplaats GN. *Numerical Optimization Techniques for Engineering Design: With Application*, McGraw-Hill, 2nd edition, New York, 1984.
16. Gholizadeh S. Layout optimization of truss structures by hybridizing cellular automata and particle swarm optimization, *Computers & Structures*, **125**(2013) 86-99.
17. Kaveh A, Mahdavi VR. Two-dimensional colliding bodies algorithm for optimal design of truss structures, *Advances in Engineering Software*, **83**(2015) 70-9.
18. Wasserman PD. *Advanced Methods in Neural Computing*, Prentice Hall Company, New York, 1993.
19. Hagan MT, Menhaj M. Training feed-forward networks with the Marquardt algorithm, *IEEE Transaction on Neural Networks*, **5**(1999) 989-93.
20. Salajegheh E, Heidari A. Optimum design of structures against earthquake by wavelet neural network and filter banks, *Earthquake Engineering & Structural Dynamics*, **34**(2005) 67-82.
21. Iranian Code of Practice for Seismic Resistant Design of Buildings (Standard 2800), 3rd Edition, BHRC Publication, No. S-465, 2007.
22. Mazzoni S, McKenna F, Scott MH, Fenves GL. *OpenSees Command Language Manual*, Pacific Earthquake Engineering Research (PEER) Center, 2006.
23. The Language of Technical Computing. *MATLAB*, Math Works Inc., 2013.
24. Structural analysis program, *SAP2000*, Computers and Structures, Inc., Berkeley, CA, 2011.