INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING
Int. J. Optim. Civil Eng., 2013; 3(1):85-98

OPTIMUM SHAPE DESIGN OF DOUBLE-LAYER GRIDS BY QUANTUM BEHAVED PARTICLE SWARM OPTIMIZATION AND NEURAL NETWORKS

S. Gholizadeh[*, †] P. Torkzadeh[2] and S. Jabarzadeh[2]
[1]Department of Civil Engineering, Urmia University, Urmia, Iran
[2]Department of Civil Engineering, Shahid Bahonar University of Kerman, Kerman, Iran

ABSTRACT

In this paper, a methodology is presented for optimum shape design of double-layer grids subject to gravity and earthquake loadings. The design variables are the number of divisions in two directions, the height between two layers and the cross-sectional areas of the structural elements. The objective function is the weight of the structure and the design constraints are some limitations on stress and slenderness of the elements besides the vertical displacements of the joints. To achieve the optimization task a variant of particle swarm optimization (PSO) entitled as quantum-behaved particle swarm optimization (QPSO) algorithm is employed. The computational burden of the optimization process due to performing time history analysis is very high. In order to decrease the optimization time, the radial basis function (RBF) neural networks are employed to predict the desired responses of the structures during the optimization process. The numerical results demonstrate the effectiveness of the presented methodology.

Received: 10 July 2012; Accepted: 20 December 2012

KEY WORDS: double-layer grid; optimum shape design; time history analysis; quantum-behaved particle swarm optimization; radial basis function neural network

1. INTRODUCTION

For design of many structures, the horizontal components of the earthquake are more effective and vertical component in particular situations which earthquake code suggested, are

[*]Corresponding author: S. Gholizadeh, Department of Civil Engineering, Urmia University, Urmia, Iran
[†]E-mail address: s.gholizadeh@urmia.ac.ir

considered. For example in the case of structures with large spans the effects of the vertical component cannot be neglected. Double-layer grids are the most general structures that are used for covering large spans without interior columns, as a result for designing of these structures considering vertical component is necessary.

As the dimensions and the number of structural elements of double-layer grids are usually very large, it is essential to evolve strategies for their optimal design. Optimal shape design of space structures with a large number of structural members leads to structures with less weight and cost and therefore the resulted structural systems appear very efficient.

During last years, meta-heuristic search techniques have been emerged as more powerful computational optimization tools with respect to conventional gradient based methods. Meta-heuristics make use of ideas inspired from the nature. The basic idea behind these techniques is to simulate natural phenomena, such as survival of the fittest, immune system, swarm intelligence and the cooling process of molten metals through annealing into a numerical algorithm [1]. The optimum structural design algorithms that are based on these techniques are robust and quite effective in finding the solution of discrete programming problems. There are large numbers of such meta-heuristic techniques available in the literature nowadays [2-4]. One of the most popular ones is particle swarm optimization (PSO). Recently, a novel variant of the PSO, called quantum-behaved particle swarm optimization (QPSO) algorithm was proposed by inspiring quantum mechanics [5]. The QPSO possesses good global search ability and it is a promising optimizer for complex problems. In the present work, PSO and QPSO are selected as the optimizers.

During the optimization process critical structural responses such as maximum deflection and stresses should not exceed the design codes' requirements. Optimal design of such large-scaled structures subjected to earthquake time history loading is very time consuming therefore to efficiently achieve the optimization task, it is necessary to somehow reduce the computational burden. One of the most popular techniques to simplify the complex problems and to considerably reduce the elapsed time is artificial intelligence, especially artificial neural networks. There are a huge number of publications in the field of neural networks' applications in the structural engineering. For example, in the case of space structures Kaveh and Servati [6] used neural networks to design of double-layer grids subject to gravity loading. Salajegheh and Gholizadeh [7] employed three neural networks model to optimum design of space structures. Rajasekaran [8] employed a cellular genetic algorithm and neural network to optimize cross-sectional areas of space structures with fixed geometry.

In the last years, a number of researchers employed meta-heuristics in conjunction with neural networks to perform optimization task of complex structures for earthquake loading spending a reasonable amount of time. Salajegheh *et al* [9] employed a binary model of PSO and neural networks to optimize truss structures for earthquake induced loads. Gholizadeh and Samavati [10] used wavelet transform and neural networks to decrease the computational burden of structural time history analysis for earthquake loading. They used an improved genetic algorithm (GA) to achieve optimization task. Gholizadeh and Salajegheh [11-12] integrated advanced neural network models with GA and PSO based meta-heuristic algorithms to fulfill structural optimization for earthquake loading.

In this study, the PSO and QPSO with discrete variables are used for shape optimization of double-layer grids subject to vertical component of earthquake loading. The radial basis

function (RBF) neural network [13] is also used to predict the dynamic responses of the structures. The design variables of the optimization are the number of divisions in two directions and the height between two layers besides the cross-sectional area of the structural elements. The numerical results demonstrate that the computational performance of the QPSO is better than that of the PSO and also by using the neural networks the time of optimization can be considerably decreased.

2. FORMULATION OF THE OPTIMIZATION PROBLEM

The optimum shape design of double-layer grids subjected to earthquake loading is to find the optimum values for cross-sectional area of the elements, the number of the divisions in two directions and the height between two layers. General formulation of the optimization problem can be presented as follows:

$$\textit{Minimize}: \quad W(X) = \sum_{k=1}^{ne} \rho_k A_k l_k \tag{1}$$

$$\textit{Subject to}: \quad g_j(X) \le 0 \ , \ j = 1,2,\ldots,m \tag{2}$$

$$X_i \in R^d \ , \ i = 1,2,\ldots,n \tag{3}$$

$$X_i = \{A_1, A_2, ..., A_{ne}, N, H\}^{\mathrm{T}} \tag{4}$$

where $W(X)$ represents the objective function, ρ_k , A_k and l_k are density, cross sectional area and length of the kth element, respectively and ne is the number of the structural elements; $g(X)$ is the design constraint; m and n are the number of the constraint and the design variables, respectively. A given set of discrete values is expressed by R^d and design variables X_i can take values only from this set. N is the number of divisions in x and y directions and H is the height between two layers.

The constraints of the optimization problem are handled by using the concept of the penalty function as follows:

$$\sigma_k < \sigma_{max} \rightarrow \phi_\sigma^k = \frac{\sigma_k - \sigma_k^{max}}{\sigma_k^{max}} \quad , \qquad k = 1,2,...,ne \tag{5}$$

$$\lambda_k < \lambda_{max} \rightarrow \phi_\lambda^k = \frac{\lambda_k - \lambda_k^{max}}{\lambda_k^{max}} \quad , \qquad k = 1,2,...,ne \tag{6}$$

$$\delta_i < \delta_{max} \rightarrow \phi_\delta^i = \frac{\delta_i - \delta_i^{max}}{\delta_i^{max}} \quad , \qquad i = 1,2,...,nj \tag{7}$$

$$\Phi = W(1+(\sum_{k=1}^{ne}(\phi_\sigma^k+\phi_\lambda^k)+\sum_{i=1}^{nj}\phi_\delta^i)^2) \tag{8}$$

where σ_k and σ_k^{max} are the current and the maximum allowable stress in *k*th element respectively; λ_k and λ_k^{max} are the current and the maximum value of the slenderness in the *k*th element; δ_k and δ_k^{max} are the current and the maximum allowable displacement in *i*th joint of the structure respectively; Φ is the pseudo objective function; ϕ_σ^k, ϕ_λ^k and ϕ_δ^i are the stress, slenderness and nodal deflection penalties, respectively.

3. PARTICLE SWARM OPTIMIZATION ALGORITHM

In this section, at first the fundamental concepts of PSO is generally described then the concepts of the QPSO are explained.

3.1 Classical Particle Swarm Optimization

PSO is a population based optimization algorithm. The population in PSO is called swarm and each individual in population of PSO is called particle. Each particle of the swarm represents a potential solution of the optimization problem. This optimization method first presented by Kennedy and Eberhart [14]. The particles fly through the search space and their positions updated based on the best positions of individual particles in each iteration. Using following equations the position of the particles is updated:

$$V_i^{k+1} = \omega V_i^k + c_1 r_1 (P_{best,i}^k - X_i^k) + c_2 r_2 (G_{best}^k - X_i^k) \tag{9}$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \tag{10}$$

where X_i and V_i represent the current position and the velocity of the *i*th particle respectively; $P_{\text{best},i}^k$ is the best previous position of the *i*th particle (called *pbest*) and G_{best}^k is the best global position among all the particle in the swarm (called *gbest*); Positive constants c_1 and c_2 are the cognitive and social components, respectively; r_1 and r_2 are two uniform random sequences generated from range (0,1) and ω is the inertia weight used discount the previous velocity of the particle persevered.

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

$$\omega = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{k_{\max}}.k \tag{11}$$

where ω_{max} and ω_{min} are the maximum and minimum values of ω, respectively. Also, k_{max}, and k are the number of maximum iterations and the number of present iteration, respectively.

3.2 Quantum Particle Swarm Optimization

In the recent decades, novel optimization methods have been proposed based on the concepts of quantum mechanics [15–17]. QPSO is one of the novel optimization methods based on quantum mechanics. In term of classical mechanics, a particle is depicted by position and velocity vectors which determine the trajectory of the particle. The particle moves along the determined trajectory in Newtonian mechanics but this is not in the quantum mechanics. In quantum world, the term trajectory is meaningless, because X_i and V_i of a particle cannot be determined simultaneously according to the uncertainty principle. Therefore if the individual particles in a PSO system have a quantum behavior, the PSO algorithm is bound to work in different fashion [18-19]. In QPSO algorithm, the position of a particle cannot be determined according to uncertainty principle. In this case, the state of a particle is depicted by wave function $\psi(x,t)$, instead of velocity and position. Thus in QPSO the particle can only learn the probability of its appearance in position X from probability density function ($|\psi(x,t)|^2$), the form of which depends on the potential field the particle lies in. Employing the Monte Carlo method, the particles move according to the following equation [18-20]:

$$\begin{cases} X_i^{k+1} = P_i + \beta \left| M_{\text{best}} - X_i^k \right| ln(1/u) & if \quad \gamma \geq 0.5 \\ X_i^{k+1} = P_i - \beta \left| M_{\text{best}} - X_i^k \right| ln(1/u) & if \quad \gamma < 0.5 \end{cases} \tag{12}$$

where P_i is the local attractor of ith particle; β is a design parameter called contraction–expansion coefficient [21]; u and γ are values generated using the uniform probability distribution functions in the range [0, 1].

The global point called Mainstream Thought or Mean Best (M_{best}) of the population is defined as the mean of the *pbest* positions of all particles and is given as follows:

$$M_{best} = \frac{1}{np} \sum_{i=1}^{np} P_{\text{best},i} \tag{13}$$

where *np* is the number of the particles in the swarm.

According to the convergence analysis of the original PSO, the convergence of the PSO algorithm may be achieved if each particle converges to its local attractor [21]. The local attractor to guarantee convergence of the PSO is presented as follows:

$$P_i = \frac{c_1 r_1}{c_1 r_1 + c_2 r_2} P_{\text{best},i} + \frac{c_2 r_2}{c_1 r_1 + c_2 r_2} G_{\text{best}} \tag{14}$$

The flowchart of the QPSO is shown in Figure 1.

Despite the efficiency of the QPSO compared with PSO, the optimization process requires a great number of time history analyses; thus the overall time of the optimization process is still very long. In this paper to reduce the computational burden, RBF neural network is employed. As the number of divisions in two directions (N) and the height between two layers (H) are also variables, a single RBF may not properly predict the

structural responses. Thus in this paper, a set of RBF neural networks are employed. In the next section the RBF neural network is generally explained.

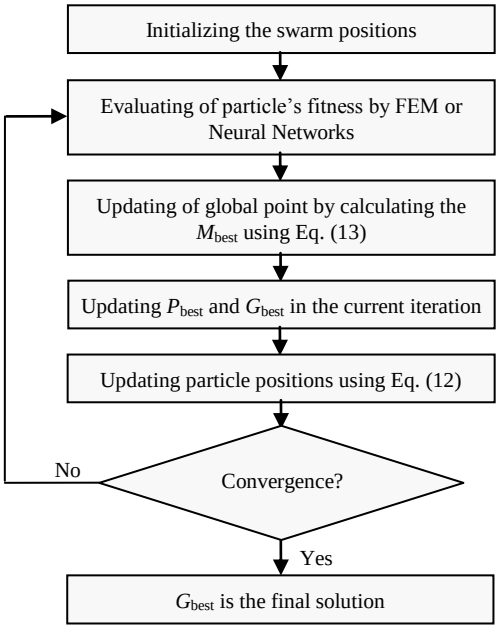

Figure 1. Flowchart of the QPSO

4. RADIAL BASIS FUNCTION NEURAL NETWORKS

One of the most popular neural network techniques is RBF due to its fast training, generality and simplicity [13]. 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 called the receptive field. The radius of the receptive field allows the sensitivity of the RBF neurons to be adjusted. During the training, the receptive field radius of RBF neurons is such determined that the neurons could cover the input space.

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 [13].

$$L_1^W = \Pi^{\mathrm{T}} \tag{15}$$

where L_1^W and Π^{T} are the 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

$$L_2^W = O^{-1} T \tag{16}$$

T is the target matrix, O is the outputs of the hidden layer and L_2^W is the output layer weight matrix.

A successful application of RBF in the case of double-layer grids was reported in [22].

5. NUMERICAL RESULTS

In this paper, square-on-square double-layer grids with the span length of 20 m are considered. The structure is subjected to the gravity loading and the vertical component of Tabas earthquake (1978) record, shown in Figure 2.

The number of divisions in both directions, N, varies between 4 and 7. The height between the top and bottom layers, H, is varied between 0.7 m and 1.9 m by the increment of 0.4 m. Each of the top, web and bottom layers is divided into three groups. For example the grouping details corresponding to 7 divisions are shown in Figure 3.

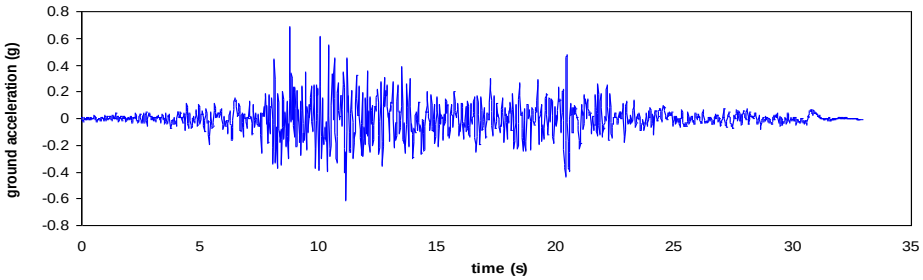

Figure 2. Vertical component of the Tabas earthquake (1978)

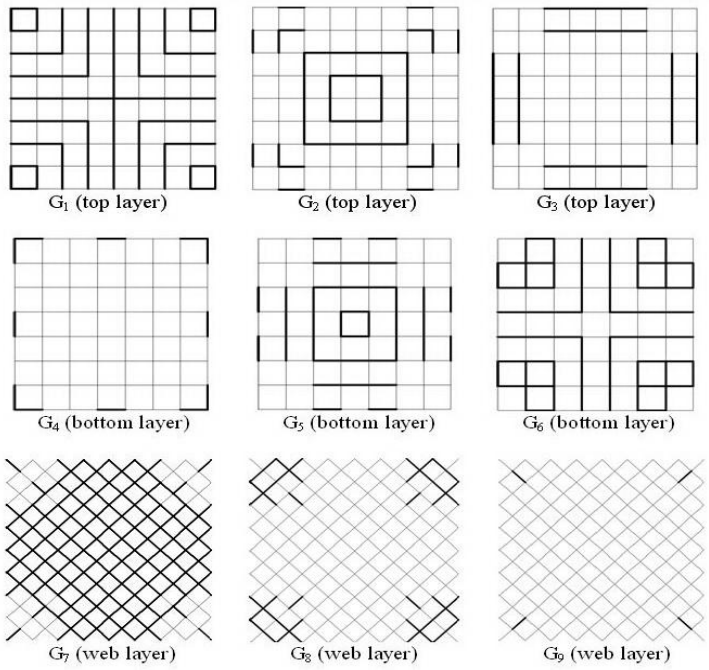

Figure 3. Grouping details for 7 divisions

The similar patterns exist for divisions of 4 to 6 but those are neglected to show here.

The cross sections of the element groups can be chosen from the pipe sections given in Table 1. In this table, *d* and *t* are the external diameter and thickness of the pipe sections, respectively.

Table 1: Available cross-sectional areas

NO.	*d (cm)*	*t (cm)*	*NO.*	*d (cm)*	*t (cm)*
1	8.89	0.32	11	21.91	0.59
2	10.16	0.40	12	21.91	0.63
3	11.43	0.50	13	21.91	0.71
4	11.43	0.56	14	21.91	0.88
5	13.97	0.56	15	24.45	0.80
6	13.97	0.63	16	24.45	1.00
7	15.90	0.63	17	24.45	1.10
8	16.83	0.50	18	32.39	0.71
9	19.37	0.63	19	32.39	0.88
10	21.91	0.50	20	35.56	1.00

The structure is simply supported at the corner nodes of the bottom layer. Young modulus is 2.1×10^6 kg/cm^2 and mass density is 7850 kg/m^3. The aim is to achieve the optimum shape and size design of the structure subjected to uniformly distributed gravity load of 150 kg/m^2 on the top layer and the Tabas earthquake (1978) record, shown in Fig 2, applied in the vertical direction.

The allowable absolute values of tension stress, compressive stress and slenderness of the elements are 1400.0 kg/cm^2, 1000.0 kg/cm^2, and 200.0, respectively. The allowable deflection is also considered 5.0 cm.

In order to reduce the computational time of shape and sizing optimization problem, a set of 16 RBF neural networks are trained to predict the necessary structural responses during the optimization process. In this case, for each number of divisions four RBF neural network model corresponding to the heights of 0.7 m, 1.1 m, 1.5 m and 1.9 m are trained. The input and output of the RBF neural networks given by Eqs (17) and (18) are the vector of cross sections of groups 1 to 9 and the vector of maximum stresses of each element groups and maximum deflection, respectively.

$$I_{\mathrm{RBF}} = \left\{ cs_{\mathrm{G1}} \quad cs_{\mathrm{G2}} \quad \ldots \quad cs_{\mathrm{G9}} \right\}^{\mathrm{T}} \tag{17}$$

$$O_{\mathrm{RBF}} = \left\{ \sigma_{G1}^{\max} \quad \sigma_{G2}^{\max} \quad \ldots \quad \sigma_{G9}^{\max} \quad \delta^{\max} \right\}^{\mathrm{T}} \tag{18}$$

The set of RBF neural networks employed during the optimization process are shown in Figure 4.

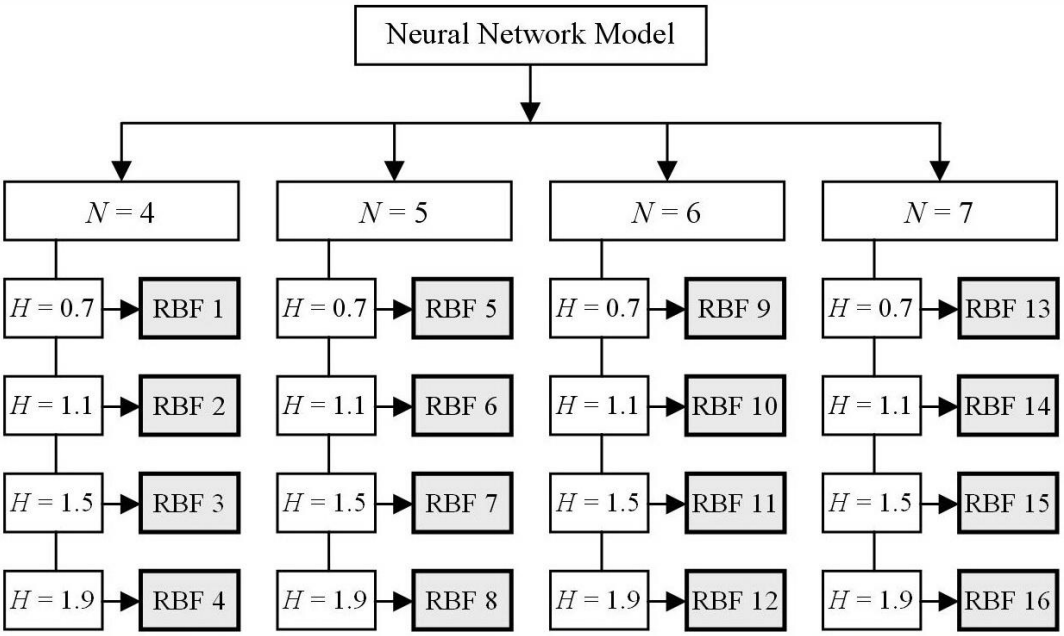

Figure 4. Neural network model comprising 16 RBF neural networks

To train each of the RBF neural networks, a number of 100 samples are generated and are analyzed for the mentioned static and dynamic loadings by FEM. In this sequence 1600 exact analysis are performed with overall computational time of 386.0 min. For each RBF neural network 80 and 20 samples are used to achieve training and testing tasks, respectively. The errors between exact and approximate deflections (e_δ) and stresses (e_σ) are computed as follows:

$$e_\delta = \frac{\left| \delta^{\max}_{\text{FEM}} - \delta^{\max}_{\text{RBF}} \right|}{\left| \delta^{\max}_{\text{FEM}} \right|} \times 100 \tag{19}$$

$$e_{\sigma,i} = \frac{\left| \sigma^{\max}_{Gi\ \text{FEM}} - \sigma^{\max}_{Gi\ \text{RBF}} \right|}{\left| \sigma^{\max}_{Gi\ \text{FEM}} \right|} \times 100 \quad , \quad i = 1,2,...,9 \tag{20}$$

The results of testing are given in Table 2 in terms of mean of error percentage of the test data. The results of this table indicate that the performance generality of the trained RBF neural networks is acceptable and those can be used during the optimization process to predict the analysis results. The optimization results of the double-layer grid are given in Table 3. The convergence histories of the optimization processes are also shown in Figure 5.

Table 2: Testing results of the 16 RBF neural networks in terms of mean error percentage

Neural Network	*Mean of errors (%)*										*Ave.*
	$e_{\sigma,1}$	$e_{\sigma,2}$	$e_{\sigma,3}$	$e_{\sigma,4}$	$e_{\sigma,5}$	$e_{\sigma,6}$	$e_{\sigma,7}$	$e_{\sigma,8}$	$e_{\sigma,9}$	e_{δ}	
RBF 1	13.9	12.0	5.4	9.9	7.1	6.1	6.3	4.1	4.2	7.5	7.7
RBF 2	11.1	7.9	4.1	7.1	6.8	5.2	6.1	4.5	4.4	6.2	6.3
RBF 3	11.3	11.1	4.6	9.6	10.1	6.1	8.4	6.7	3.7	5.1	7.7
RBF 4	15.4	14.5	5.5	13.1	9.3	4.5	8.1	7.8	4.9	8.4	9.2
RBF 5	10.1	8.2	6.7	10.1	7.7	7.2	8.3	9.9	5.1	8.3	8.2
RBF 6	10.8	9.5	3.7	8.4	7.1	6.9	6.9	5.6	4.2	6.1	6.9
RBF 7	9.3	7.5	4.8	7.8	6.6	5.9	9.1	8.9	3.5	5.4	6.9
RBF 8	13.8	8.6	6.1	10.3	8.6	5.9	8.1	8.4	3.6	5.4	7.9
RBF 9	9.8	9.7	5.5	7.5	9.5	5.2	8.9	10.3	6.9	10.5	8.4
RBF 10	9.1	8.4	3.5	5.4	8.7	4.1	6.9	6.9	4.0	5.4	6.2
RBF 11	11.6	9.2	3.4	6.4	6.9	4.6	7.1	7.8	3.4	7.5	6.8
RBF 12	10.1	10.1	5.9	10.2	7.2	5.9	8.9	7.2	4.9	7.5	7.8
RBF 13	9.8	7.3	4.9	7.7	6.6	5.1	8.2	6.8	4.8	5.4	6.7
RBF 14	9.9	8.4	5.7	9.9	10.3	5.4	11.1	9.9	5.9	5.2	8.2
RBF 15	12.1	6.3	4.1	6.4	7.4	4.2	7.3	6.7	3.5	6.6	6.5
RBF 16	9.7	8.2	5.4	12.3	6.5	5.9	7.6	7.3	4.4	4.6	7.2

The results demonstrate that QPSO is superior to PSO in terms of optimal weight and the number of required iterations.

Table 3: Optimum design results of various methods (d×t)

Design variables	*PSO*		*QPSO*	
	RBF	FEM	RBF	FEM
cs_{G1}	8.89×0.32	8.89×0.32	8.89×0.32	8.89×0.32
cs_{G2}	11.43×0.50	11.43×0.56	11.43×0.50	11.43×0.50
cs_{G3}	16.83×0.50	19.37×0.63	13.97×0.63	16.83×0.50
cs_{G4}	8.89×0.32	8.89×0.32	8.89×0.32	8.89×0.32
cs_{G5}	8.89×0.32	8.89×0.32	8.89×0.32	8.89×0.32
cs_{G6}	13.97×0.56	16.83×0.50	13.97×0.56	13.97×0.56
cs_{G7}	8.89×0.32	8.89×0.32	8.89×0.32	8.89×0.32
cs_{G8}	8.89×0.32	11.43×0.50	8.89×0.32	10.16×0.40
cs_{G9}	19.37×0.63	21.91×0.59	13.97×0.56	19.37×0.63
N	7	7	7	7
H (m)	1.90	1.90	1.90	1.90
Weight (kg)	12681.14	14149.43	12323.54	13030.53
Iteration	126	127	57	78
Training time (min.)	386.0	-	386.0	-
Optimization time (min.)	2.9	1509.1	1.4	932.6
Overall time (min.)	388.9	1509.1	387.4	932.6

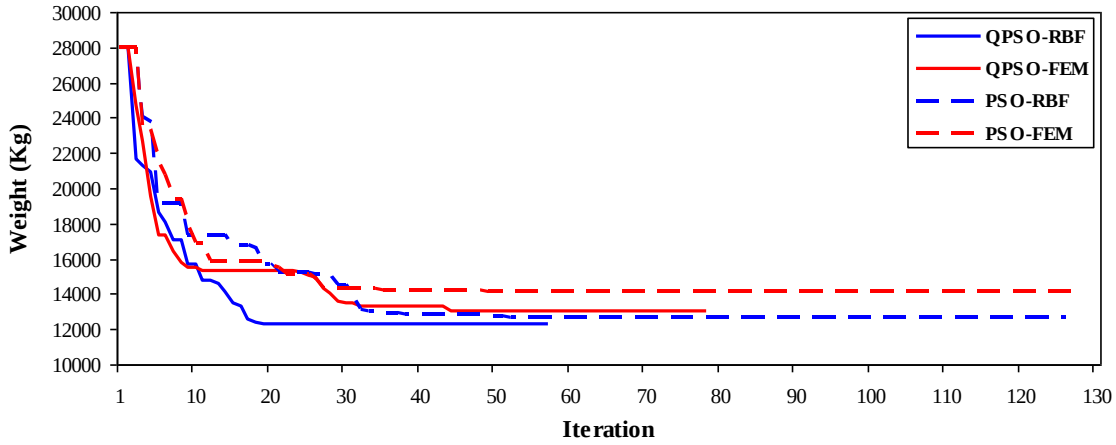

Figure 5. Convergence histories of various optimization methods

Stress ratios of the optimum designs found by PSO and QPSO using FEM are shown in Figures 6 and 7, respectively. The maximum deflections of these optimum designs are also 4.88 cm and 4.95, respectively which are less than their allowable values.

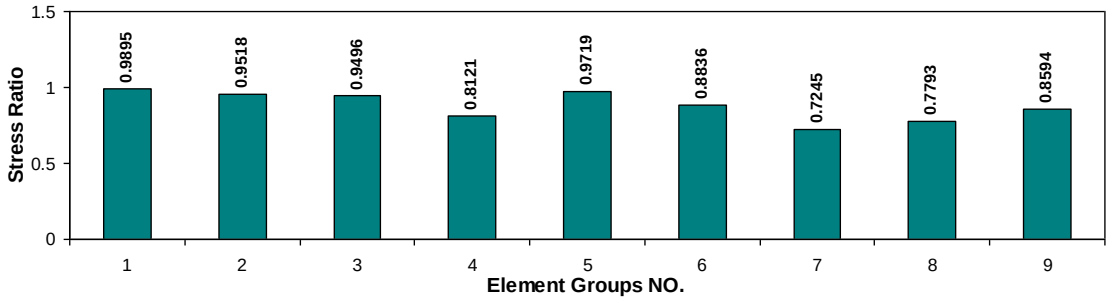

Figure 6. Stress ratios of the optimum design found by PSO using FEM

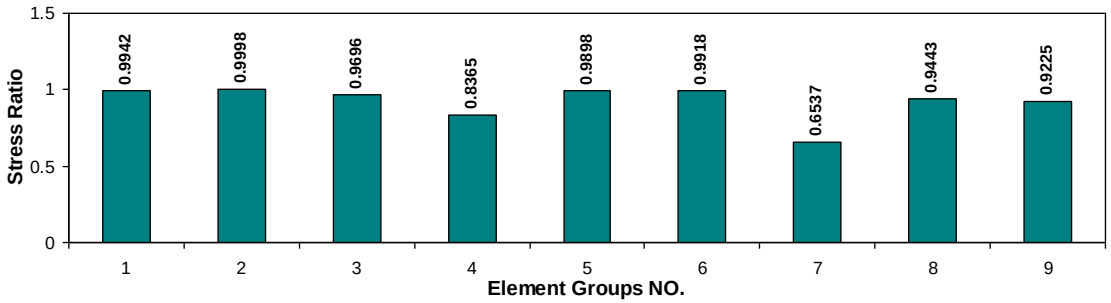

Figure 7. Stress ratios of the optimum design found by QPSO using FEM

To assess the feasibility of the solutions found by PSO and QPSO using RBFs the final solutions are analyzed by FEM and their element stress ratios and maximum deflections are computed. These computed values are denoted as actual stress ratios. The stress ratios

predicted by RBFs are compared with their corresponding actual values in Figures 8 and 9, respectively. The maximum deflections of these optimum designs evaluated by RBF16 are 4.41 and 4.31 cm and their actual values evaluated by FEM are also 5.03 cm and 4.98, respectively.

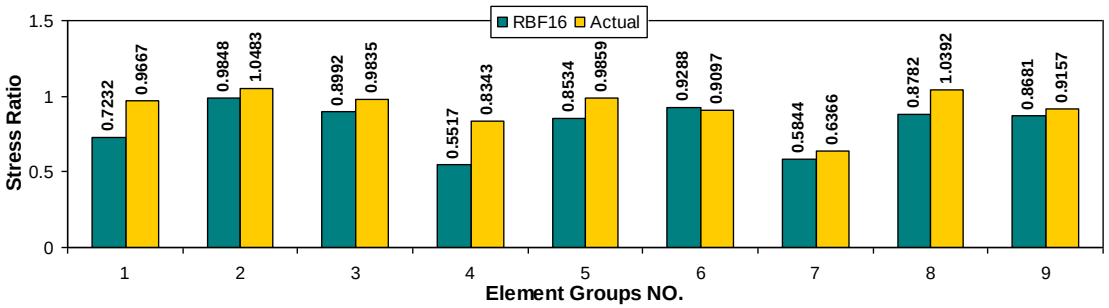

Figure 8. Stress ratios of the optimum design found by PSO using RBFs

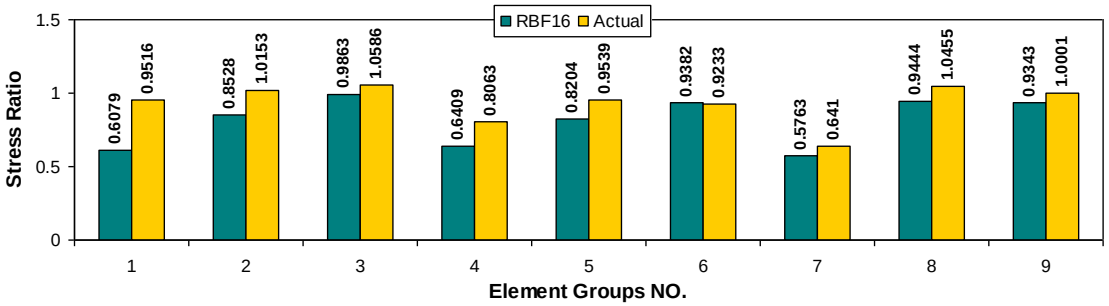

Figure 9. Stress ratios of the optimum design found by QPSO using RBFs

The results reveal that however all of the stress ratios predicted by RBF16 are less than 1.0 but in some of the element groups the corresponding actual values are slightly greater than 1.0. This is a numerical problem which can be totally removed by improving the performance generality of the neural network model using more training samples.

6. CONCLUSIONS

In this paper, a combination of meta-heuristic optimization algorithms and neural networks are employed for optimum shape design of double-layer grids subject to gravity and earthquake loadings. PSO and QPSO meta-heuristics algorithms are used for achieving the optimization task. The computational burden of the optimization due to performing time history analysis is generally high. In order to mitigate this computational rigor, RBF neural networks are trained to predict the structural responses required during the optimization process. Due to this fact that the number of divisions and the height between two layers are treated as design variables, a single RBF cannot efficiently predict the structural responses. In this study a set of RBF neural networks are employed to achieve the prediction task. The optimization results show that the QPSO algorithm incorporating RBF neural networks,

denoted as QPSO-RBF, converges to optimum design with a weight of 12323.54 kg by 57 generations which is better than 12681.14 kg associated with the PSO-RBF by 126 generations. It is also observed that the QPSO-FEM find the optimal weight of 13030.53 kg with 78 generation which is better than 14149.43 kg of PSO-FEM with 127 generations. These results demonstrate that QPSO converges to better solutions with a lower computational cost compared with PSO. The overall computational time of optimization by PSO using RBF neural networks, including data generation and training times, is about 0.26 times the time required by PSO using FEM. In the case of QPSO this ratio is about 0.4. It can be therefore concluded that the complex problem of double-layer grids shape optimization subject to earthquake loading can be efficiently tackled by QPSO meta-heuristic using RBF neural networks spending low computational cost.

REFERENCES

1. Hasancebi O, Carbas S, Dogan E, Erdal F, Saka MP. Performance evaluation of metaheuristic search techniques in the optimum design of real size pin jointed structures, *Comput Struct*, 2009; **87**: 284–302.
2. Kaveh A, Zakian P. Performance based optimal seismic design of RC shear walls incorporating soil-structure interaction using CSS algorithm, *Int J Optim Civil Eng*, 2011; **2**: 383–405.
3. Gholizadeh S, Barati H. A comprative study of three metaheurustics for optimum design of trusses, *Int J Opt Civil Eng*, 2012; **3**: 423–41.
4. Kazemzadeh Azad S, Kazemzadeh Azad S. Optimum design of structures using an improved firefly algorithm, *Int J Optim Civil Eng*, 2011; **2**: 327–40.
5. Sun J, Feng B, Xu WB. Particle swarm optimization with particles having quantum behavior. *Proceedings of 2004 Congress on Evolutionary Computation*, 2004, pp. 326–331.
6. Kaveh A, Servati H. Design of double-layer grids using back propagation neural networks, *Comput Struct*, 2001; **79**: 1561–8.
7. Salajegheh E, Gholizadeh S. Optimum design of structures by an improved genetic algorithm using neural networks, *Adv Eng Soft*, 2005; **36**: 757–67.
8. Rajasekaran S. Optimization of large scale three dimensional reticulated structures using cellular genetics and neural networks, *Int J Space Struct*, 2001; **16**: 315–24.
9. Salajegheh E, Gholizadeh S, Khatibinia M. Optimal design of structures for earthquake loads by a hybrid RBF-BPSO method, *Earth Eng Eng Vib*, 2008; **7**: 13–24.
10. Gholizadeh S, Samavati OA. Structural optimization by wavelet transforms and neural networks, *Appl Math Model*, 2011; **35**: 915-29.
11. Gholizadeh S, Salajegheh E. Optimal seismic design of steel structures by an efficient soft computing based algorithm, *J Constr Steel Res*, 2010; **66**: 85–95.
12. Gholizadeh S, Salajegheh E. Optimal design of structures for time history loading by swarm intelligence and an advanced met model, *Compu Meth Appl Mech Eng*, 2009; **198**: 2936–49.
13. Wasserman PD. *Advanced Methods in Neural Computing*, Prentice Hall Company, Van

Nostrand Reinhold, New York, 1993.
14. Kennedy J, Eberhart RC. Particle swarm optimization, *Proceedings of the IEEE International Conference on Neural Networks*, Perth, Australia, 1995, pp. 1942–1945.
15. Bulger D, Baritompa WP, Wood GR. Implementing pure adaptive search with Grover's quantum algorithm, *J Opt Theory Appl*, 2003; **116**: 517–29.
16. Han KH, Kim JH. Quantum-inspired evolutionary algorithm for a class of combinatorial optimization, *IEEE Trans Evol Comput*, 2002; **6**: 580–93.
17. Wang L, Tang F, Wu H. Hybrid genetic algorithm based on quantum computing for numerical optimization and parameter estimation, *Appl Math Comput*, 2005; **171**: 1141–56.
18. Sun J, Feng B, Xu W. Particle swarm optimization with particles having quantum behavior, *Proceedings of Congress on Evolutionary Computation*, 2004, 325-331.
19. Sun J, Xu W, Feng B. Adaptive parameter control for quantum-behaved particle swarm optimization on individual level, *Proceedings of IEEE International Conference on Systems*, Man and Cybernetics, Big Island (HI, USA), 2005, pp. 3049-3054.
20. Xi M, Sun J, Xu W. An improved quantum-behaved particle swarm optimization algorithm with weighted mean best position, *Appl Math Comput*, 2005; **205**: 751–9.
21. Clerc M, Kennedy JF. The particle swarm: Explosion, stability and convergence in a multi-dimensional complex space, *IEEE Trans Evol Comput*, 2002; **6**, 58–73.
22. Gholizadeh S, Sheidaii MR, Farajzadeh S. Seismic design of double layer grids by neural networks, *Int J Opt Civil Eng*, 2012; **2**: 29–45.