ASIAN JOURNAL OF CIVIL ENGINEERING (BUILDING AND HOUSING) VOL. 13, NO. 1 (2012)
PAGES 79-88

RELIABILITY ASSESSMENT OF STRUCTURES BY MONTE CARLO SIMULATION AND NEURAL NETWORKS

R. Kamyab Moghadas[a*] and M.J. Fadaee[b]
[a]The Iranian Academic Center for Education, Culture and Research, and Kerman, Iran
[b]Civil Engineering Department, Shahid Bahonar University of Kerman, Kerman, Iran

Received: 5 June 2010, **Accepted:** 25 April 2011

ABSTRACT

This paper examines a methodology for computing the probability of structural failure by combining Monte Carlo Simulation (MCS) and Artificial Neural Networks (ANN). MCS is a powerful tool, simple to implement and capable of solving a broad range of reliability problems. However, its use for evaluation of very low probabilities of failure implies a great number of structural analyses which can become excessively time consuming. In the present study, nonlinear structural analysis is involved and therefore the computational effort of MCS will be resonated comparing with that of the linear analysis. The proposed methodology makes use of capability of a ANN to approximate a function for reproducing structural behavior, allowing the computation of performance measures at a much lower cost. In order to assess the validity of this methodology, a structural example is presented and discussed. The numerical results demonstrate the efficiency of the proposed methodology for the structural reliability analysis.

Keywords: Structural reliability; Monte Carlo simulation; nonlinear analysis; neural networks

1. INTRODUCTION

The structural designer must verify, within a prescribed safety level, the serviceability and ultimate conditions commonly expressed by the inequality: $S_d < R_d$, where S_d represents the action effect and R_d the resistance. The intrinsic random nature of material properties and actions is actually considered by some codes. In the present study a more accurate and also computationally efficient method is employed to deal with this randomness. In this method the probability of failure is computed from the joint probability distribution of the random variables associated with the actions and resistances.

Theory and methods for structural reliability have been developed substantially in the last

* E-mail address of the corresponding author: r_kamyab_m@yahoo.com (R. Kamyab Moghadas)

few years and they are actually useful tools for evaluating rationally the safety of complex structures. Recent developments allow anticipating that their application will gradually increase, even in the case of common structures.

Monte Carlo Simulation (MCS) is a simulation method that presents the following characteristics: it can be applied to many practical problems allowing the direct consideration of any type of probability distribution for the random variables; it is able to compute the probability of failure with the desired precision; and it is easy to implement. However, despite the advantages it presents, the use of this method is not widespread in structural reliability because it is not efficient in terms of computational burden. In fact, MCS requires a great number of structural analyses, one for each sample of the set of random variables. The number of analyses needed to evaluate the probability of failure of a structure with a prescribed precision depends on the order of magnitude of that probability. As the values of the probability of failure associated to the ultimate limit states vary normally between 10^{-4} and 10^{-6}, for ensuring a 95% likelihood that the actual probability be within 5% of the computed one, the number of analyses to be performed must be at least 1.6×10^{7} to 1.6×10^{9}, according to Shooman [1].

These analyses are frequently performed with the help of finite elements software. Therefore, the computation time can be prohibitively high, especially when the structure exhibits non-linear behavior or the numerical model is rather complex. To eliminate this drawback, it is proposed here the use of Neural Networks (NN) to approximate structural response. Once properly trained, an NN allows the determination of the structural performances with a very small number of operations and at a fraction of the cost of the corresponding structural analysis. This methodology allows the application of MCS to practical cases of great complexity where the direct use of this method would not be feasible.

To examine the computational performance of the proposed methodology a test example is presented. In this example, the reliability analysis of a steel plane truss considering nonlinear behavior is performed. Moreover, by changing the cross-sectional areas vector of the structure, the failure probability of the structure is computed employing MCS coupled with back-propagation NN. In this manner a database including 50 samples are provided. At last, by using the database, a back-propagation NN is trained to predict the probability of failure of the structure. In this NN, the input is the cross-sectional areas vector and the output is the probability of failure. The numerical results show a good agreement between the predicted and evaluated values of the probability of failure. Through this test example, it is demonstrated that the proposed methodology is a robust tool for reliability analysis of the structures.

2. THEORETICAL BACKGROUND OF NONLINEAR ANALYSIS

In a linear analysis we implicitly assume that the deflections and strains are very small and the stresses are smaller than the material yield stresses. Consequently, the stiffness can be considered to remain constant and the finite element equilibrium equations are linear.

In many structures, at or near failure (ultimate) loads, the deflections and the stresses do not change proportionally with the loads. Either the stresses are so high that they no longer obey Hooke's law or else there are such large deflections that the compatibility equations cease to be linear. These two conditions are called material nonlinearity and geometric nonlinearity, respectively. In this study, a finite elements model based on geometrical and material nonlinear analysis of 2-D trusses including plasticity, and large deflection capabilities is presented by ANSYS [2]. In elasto-plastic analysis the von mises yield function is used as yield criterion. Flow rule in this model is associative and the hardening rule is bilinear isotropic hardening. In the bilinear model the slope of the second line is chosen to be zero.

2.1 Nonlinear Analysis Combining Geometrical and Material Nonlinearities

Here, instead of the linear strain-displacement relation, the nonlinear Green's strain [3] is used which is defined as follows:

$$\varepsilon_G = \frac{l_n^2 - l_0^2}{2l_0^2} \tag{1}$$

where ε_G is the nonlinear Green's strain, l_n and l_0 are the length of space truss element after and before deflection, respectively.

Since Green's strains are used, the stresses in each analysis, including geometric nonlinearity, will be 2nd order Piola-Kirchoff [3] stresses.

When the strains are nonlinear functions of the displacements or, in other words, when the stresses reach values exceeding the yield stresses of the material, the stress-strain relationship is nonlinear. In these cases, the stiffness is dependent on the displacements and the strains. Obviously, the solution of the displacements cannot be obtained in a single step. Instead, the analysis is carried out by the incremental method [3] combined with some iterative equilibrium corrections at every step.

3. MONTE CARLO SIMULATION (MCS)

A reliability problem is normally formulated using a failure function, $g(X_1,X_2,\ldots,X_n)$, where $X_1,X_2,\ldots,X_n$ are random variables. Violation of the limit state is defined by the condition $g(X_1,X_2,\ldots,X_n) \le 0$ and the probability of failure, p_f, is expressed by the following expression [4]:

$$p_{\mathrm{f}} = P[g(X_1, X_2,\ldots, X_n) \le 0] = \iint_{g(X_1,X_2,\ldots,X_n)}\cdots\int f_{X_1,X_2,\ldots,X_n}(x_1, x_2,\ldots, x_n)\mathrm{d}\,x_1\,\mathrm{d}\,x_2 \ldots\mathrm{d}\,x_n \tag{2}$$

where $(x_1, x_2,\ldots, x_\mathrm{n})$ are values of the random variables and $f_{X_1,X_2,\ldots,X_n}(x_1, x_2,\ldots, x_n)$ is the joint probability density function.

The Monte Carlo method allows the determination of an estimate of the probability of failure, given by

$$\overline{p}_{\mathrm{f}} = \frac{1}{N}\sum_{i=1}^{N} I(X_1, X_2, ..., X_n) \tag{3}$$

where $I(X_1, X_2, ..., X_n)$ is a function defined as:

$$I(X_1, X_2, ..., X_n) = \begin{cases} 1 & \text{if } \; g(X_1, X_2, ..., X_n) \le 0 \\ 0 & \text{if } \; g(X_1, X_2, ..., X_n) > 0 \end{cases} \tag{4}$$

According to (2), N independent sets of values $x_1, x_2, ..., x_n$ are obtained based on the probability distribution for each random variable and the failure function is computed for each sample. Using MCS, an estimate of the probability of structural failure is obtained by

$$\overline{p}_{\mathrm{f}} = \frac{N_{\mathrm{H}}}{N} \tag{5}$$

where N_{H} is the total number of cases where failure has occurred.

4. ARTIFICIAL NEURAL NETWORKS

NN are numerical algorithms inspired in the functioning of biological neurons. This concept was introduced by McCulloch and Pitts [5], who proposed a mathematical model to simulate neuron behavior. Use of NN has become widespread in several fields of engineering, such as structural mechanics [6-15] and structural reliability [16].

In this study back-propagation NN is employed. In the next subsection the theoretical background of the back-propagation NN is briefly explained.

4.1 Back-propagation Neural Network

The most popular and successful learning method for training the multilayer neural networks is the back-propagation algorithm. The algorithm employs an iterative gradient-descent method of minimization which minimizes the mean squared error between the desired output and the network output (supervised learning). The back-propagation training procedure is presented as,

$$E = \frac{1}{2}\sum_{j=1}^{N}\sum_{i=1}^{M} e_i^2(n) \tag{6}$$

in which N, M, and n are number of training input patterns, dimension of output space, and number of iterations, respectively, and

$$e_i(n) = d_i(n) - y_i^{(L)}(n) \tag{7}$$

where $d_i(n)$ is the desired output, $y_i^{(L)}(n)$ is the network output and L is the output layer.

The output of layer l, $v_i^l(n)$, is defined as follows,

$$v_i^l(n) = \sum_{j=1}^{N} w_{ij}^{(l)} y_j^{(l-1)}(n) \tag{8}$$

in which $y_j^{(l-1)}(n)$ is the function signal of neuron j in the previous layer, $l-1$, at iteration n, $w_{ij}^{(l)}(n)$ is the weight of neuron i in layer l that is fed from neuron j in layer $l-1$.

Then the output signal of neuron i in layer l is

$$y_i^l(n) = f(v_i^l(n)) \tag{9}$$

where $f(.)$ is the activation function.

If neuron i, is in the first hidden layer, ($l = 1$), then set $y_i^0(n)$ is $x_i(n)$

The local error or the local gradient is defined as:

$$\delta_i(n) = -\frac{\partial E_i}{\partial v_i} \tag{10}$$

Equation (10) can be simplified to

for neuron i in output layer L:

$$\delta_i^L(n) = e_i^L(n) f'(v_i^L(n)) \tag{11}$$

for neuron i in hidden layer l:

$$\delta_i^l(n) = f'(v_i^l(n)) \sum_k \delta_k^{(l+1)}(n) w_{ki}^{(l+1)}(n) \tag{12}$$

where $f'(v_i^l(n))$ is the derivative of the activation function with respect to $v(n)$.

If the activation function is chosen to be the hyperbolic tangent function then $f'(v_i)$ is:

$$f'(v_i) = \frac{df(v_i)}{dv_i} = \gamma(1 - f^2(v_i)) \tag{13}$$

where γ is an adjusting coefficient.

Hence, adjust the weights of the network in layer l according to the generalized following delta rule:

$$w_{ij}^{(l)}(n+1) = w_{ij}^{(l)}(n) + \mu \delta_i^{(l)}(n) y_j^{(l-1)}(n) \tag{14}$$

where μ is the positive constant learning rate, usually equals 0.01.

If after updating the weights, the error E is not minimized, new iterations are required.

5. NUMERICAL EXAMPLE

The structural model involved here is introduced. The structure configuration and the element groups is shown in Figure 1.

The cross-sectional area of each element group is considered to be deterministic variable and is taken from the following 25 discrete values, i.e., 1.74, 2.27, 2.67, 3.08, 3.79, 3.90, 4.80, 5.69, 6.91, 8.13, 8.70, 9.03, 9.40, 10.10, 10.57, 11.40, 11.79, 12.20, 13.90, 15.10, 15.50, 17.10, 18.70, 19.20, and 20.30 cm^2.

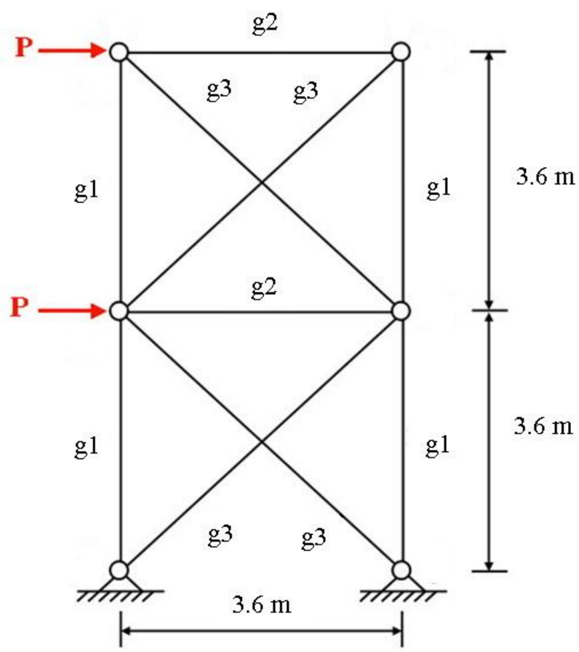

Figure 1. A 10-bar steel truss

Random variables of the truss are listed in Table 1.

Table 1: Random variables

Variable	Distribution type	Mean	Standard deviation
Modulus of elasticity (E)	Normal	21000 (KN/m^2)	1050
Yield stress (σ_y)	Normal	21 (KN/m^2)	1
External load (P)	Normal	100 (KN)	20

In this study, the failure criterion is defined as follows:

(*Ultimate Load* obtained by nonlinear analysis) < (*Applied Load*)

In the nonlinear analysis process, if due to a load increment the convergence is failed that load is called *Ultimate Load*.

In this study a hybrid methodology is employed to reliability assessment of the structure. The fundamental steps of the proposed methodology are as follows:

1. 50 sample structures with various cross-sectional areas are randomly selected.

$$A_i = \begin{bmatrix} A_{g1} \\ A_{g2} \\ A_{g3} \end{bmatrix}_i \quad , i=1,2,\ldots,50$$

2. The probability of failure of each sample structure is computed by MCS and back-propagation NN as follows:
   2.1. 750 vectors containing random variables are selected as:

$$RV_j = \begin{bmatrix} E \\ \sigma_y \\ P \end{bmatrix}_j \quad ,j=1,2,\ldots,750$$

   2.2. The ultimate loads of these 750 structures are calculated by the nonlinear analysis.
   2.3. Employing these training set, *RV*, as input and ultimate load as the output, a back-propagation NN is trained. This NN is called Local Neural Network, (LNN)
   2.4. 1.6×10^7 vectors of *RV* is selected and their corresponding ultimate load values are predicted by the trained NN. Therefore, now it is possible to calculate the probability of failure of the current structure
3. The steps 2-1 to 2-4 are repeated for all the 50 samples: $\overline{p}_{\mathrm{f}_i}$, i=1,2,…,50.
4. Now the final training set is reached: inputs: A_i ; outputs: $\overline{p}_{\mathrm{f}_i}$, i=1,2,…,50. Therefore a back-propagation NN is trained to predict the $\overline{p}_{\mathrm{f}}$ of the structures. This NN is called Global Neural Network, (GNN).

5.1 Numerical Results

For training of 50 LNN, The number of training samples is 650 while the number of testing samples is 100.

Information regarding the performance generality of 6 selective LNN is given in Table 2.

It can be observed that all the trained NNs possess appropriate performance generality.

The values of the probability of failure predicted by the 50 LNNs are listed in Table 3.

For training of GNN, the number of training samples is 40 while the number of testing samples is 10. The testing results of the GNN are given in Table 4.

Table 2: Performance generality of 6 selective LNN

No. of LNN	Error (%)		
	mean	**min**	**max**
1	0.0082	0.0001	0.0332
7	0.1244	0.0003	0.5263
12	0.2900	0.0031	1.2937
25	0.4451	0.0116	1.9670
34	0.7858	0.0003	3.3774
48	1.1838	0.0337	5.0815
Average values of all the 50 LNNs	0.5216	0.0093	2.5465

Table 3: The values of the probability of failure predicted by the 50 LNNs

No.	$\overline{p}_{\mathrm{f}}$	**No.**	$\overline{p}_{\mathrm{f}}$	**No.**	$\overline{p}_{\mathrm{f}}$	**No.**	$\overline{p}_{\mathrm{f}}$	**No.**	$\overline{p}_{\mathrm{f}}$
1	0.0014	11	0.0376	21	0.0184	31	0.0191	41	0.0189
2	0.0663	12	0.0185	22	0.0646	32	0.0014	42	0.0247
3	0.0013	13	0.0252	23	0.0190	33	0.0252	43	0.0015
4	0.0015	14	0.0247	24	0.0092	34	0.0386	44	0.018
5	0.0658	15	0.0089	25	0.0243	35	0.0248	45	0.0649
6	0.0378	16	0.0014	26	0.0186	36	0.0091	46	0.067
7	0.0094	17	0.0012	27	0.0655	37	0.0373	47	0.0192
8	0.0379	18	0.0647	28	0.0183	38	0.0084	48	0.0655
9	0.0090	19	0.0251	29	0.0187	39	0.0242	49	0.0383
10	0.0258	20	0.0381	30	0.0188	40	0.0093	50	0.0256

Table 4: testing results of GNN

Test Samples No.	$\overline{p}_{\mathrm{f}}^{\mathrm{exact}}$	$\overline{p}_{\mathrm{f}}^{\mathrm{approximate}}$	**Error (%)**
1	0.0191	0.0180	5.7764
2	0.0373	0.0403	8.0334
3	0.0184	0.0180	1.9482
4	0.0646	0.0569	11.8513
5	0.0378	0.0358	5.2233
6	0.0187	0.0179	4.2239
7	0.0383	0.0391	2.2437
8	0.0188	0.0177	5.7015
9	0.0243	0.0250	3.2201
10	0.0376	0.0372	0.8966
Average error (%)			4.9119

6. CONCLUSIONS

An efficient methodology is proposed to evaluate the probability of structural failure employing Monte Carlo simulation (MCS) and neural networks (NN). In the proposed methodology, extensive nonlinear structural analysis is involved to assess that whether the trail structures bear the external load or lose their stability and collapse. Thus, the computational effort of the MCS is considerably resonated comparing with the state that the linear analysis is employed and simple limit states are checked. Therefore NN is employed to mitigate the computational rigor of the process. In this study, backpropagation NN is employed in two stages. In the first stage, the NN is employed to predict the structural responses. In this case, for a structure with fixed cross-sectional areas, the MCS can be achieved by incorporating the properly trained NN with much less computational effort. In the numerical example in this paper, if NN is not used, the time spent to MCS of each structure is about 4.8×10^7 seconds, while using the NN reduces the time consumption to 2.25×10^3 seconds plus the time spent for NN training which is equal to 240 seconds. In the second stage, another NN is employed to predict the failure probability of structures by changing the cross-sectional areas. The numerical results demonstrate the computational advantages of the proposed hybrid methodology. Finally, it is observed that the computational time of the reliability assessment of the structures can be dramatically reduced by employing the proposed methodology.

REFERENCES

1. Shooman ML. *Probabilistic Reliability: An Engineering Approach*. McGraw-Hill, New York, 1968.
2. ANSYS Release 10.0 Documentation, ANSYS Incorporated, 2006.
3. Crisfield MA. *Non-Linear Finite Element Analysis of Solids and Structures*. Volume 1: Essentials, Chichester, UK, John Wiley & Sons, 1991.
4. Mahadevan S. In: Cruse TA, editor. *Monte Carlo Simulation: Reliability-based Mechanical Design*. Marcel Dekker; 1997. pp. 123–46.
5. McCulloch WS, Pitts W. A logical calculus of ideas immanent in nervous activity. *Bull Math Biophys*, **5**(1943)115–33.
6. Berke L, Hajela P. Applications of artificial neural nets in structural mechanics. *Structural Optimization*, **4**(1992)80-98.
7. Bento JP. Modeling mechanical behavior without mechanics. In: Proceedings of Development of Knowledge-Based Systems for Engineering. Springer-Verlag; 1998. pp. 37–58.
8. Rajasekaran S. Optimization of large scale three dimensional reticulated structures using cellular genetics and neural networks. *International Journal of Space Structures*, **16**(2001)315-24.
9. Kaveh A, Servati H. Design of double layer grids using backpropagation neural networks. *Computers and Structures*, **79**(2001)1561-8.
10. Kaveh A, Dehkordi MR. Neural networks for the analysis and design of domes.

International Journal of Space Structures, **18**(2003)181-93.

11. Salajegheh E and Gholizadeh S. Optimum design of structures by an improved genetic algorithm using neural networks. *Advances in Engineering Software*, **36**(2005)757-67.
12. Salajegheh E, Gholizadeh S, Pirmoz A. Self-organizing back propagation networks for predicting the moment-rotation behavior of bolted connection. *Asian Journal of Civil Engineering*, **9**(2008)629-45.
13. Salajegheh E, Gholizadeh S. 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.
14. Gholizadeh S. Structural optimization for earthquake loading with nonlinear responses by surrogate modeling based evolutionary algorithms. *Asian Journal of Civil Engineering*, **11**(2010)25-42.
15. Hurtado JE. Neural networks in stochastic mechanics. *Archives of Computational Methods in Engineering*, No. 3, **8**(2001)303–42.