ELSEVIER

Comput. Methods Appl. Mech. Engrg. 191 (2002) 3491–3507

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

Reliability-based structural optimization using neural networks and Monte Carlo simulation

Manolis Papadrakakis *, Nikos D. Lagaros

Institute of Structural Analysis and Seismic Research, National Technical University of Athens, Zografou Campus, Athens 15780, Greece

Received 9 July 2001; received in revised form 14 February 2002; accepted 14 February 2002

Abstract

This paper examines the application of neural networks (NN) to reliability-based structural optimization of large-scale structural systems. The failure of the structural system is associated with the plastic collapse. The optimization part is performed with evolution strategies, while the reliability analysis is carried out with the Monte Carlo simulation (MCS) method incorporating the importance sampling technique for the reduction of the sample size. In this study two methodologies are examined. In the first one an NN is trained to perform both the deterministic and probabilistic constraints check. In the second one only the elasto-plastic analysis phase, required by the MCS, is replaced by a neural network prediction of the structural behaviour up to collapse. The use of NN is motivated by the approximate concepts inherent in reliability analysis and the time consuming repeated analyses required by MCS.

Keywords: Structural optimization; Reliability analysis; Monte Carlo simulation; Evolution strategies; Neural networks; Parallel computations

1. Introduction

Reliability analysis methods have been developed over the last two decades [1,2] and have stimulated the interest for the probabilistic optimum design of structures. Despite the theoretical advancements in the field of reliability analysis serious computational obstacles arise when treating realistic problems. In particular, the reliability-based optimization (RBO) of large-scale structural systems is an extremely computationally intensive task, as shown by Tsompanakis and Papadrakakis [3]. Despite the improvements achieved on the efficiency of the computational methods for treating reliability analysis problems, they still require disproportionate computational effort for practical reliability problems. This is the reason why very few successful numerical investigations are known in the field of RBO [4–10].

* Corresponding author. Tel.: +30-1-772-1694; fax: +30-1-772-1693.
E-mail addresses: mpapadra@central.ntua.gr (M. Papadrakakis), nlagaros@central.ntua.gr (N.D. Lagaros).

0045-7825/02/$ - see front matter
PII: S0045-7825(02)00287-6

In the present study the reliability-based sizing optimization of multi-storey 3-D frames is investigated. The objective function is the weight of the structure while the constraints are both deterministic (stress and displacement limitations) and probabilistic (the overall probability of failure of the structure). Randomness of loads, material properties, and member geometry are taken into consideration in reliability analysis using the Monte Carlo simulation (MCS) method. The probability of failure of the frame structures is determined via a limit elasto-plastic analysis.

The optimization part is solved using evolution strategies (ES), which in most cases are more robust and present a better global behaviour than mathematical programming methods [11,12]. In this work two methodologies combining evolution strategies and neural networks (ES–NN) are examined. In the first one, a trained NN utilizing information generated from a number of properly selected design vectors, computed by conventional finite element and reliability analyses, is used to perform both deterministic and probabilistic constraints checks during the optimization process. The data obtained from these analyses are processed in order to obtain the necessary input and output pairs which are subsequently used for training the NN. The trained NN is then applied to predict the response of the structure in terms of deterministic and probabilistic constraints checks due to different sets of design variables. The NN training is considered successful when the predicted values resemble closely to the corresponding values of the conventional analyses which are considered exact. In the second methodology the limit elasto-plastic analyses required during the MCS are replaced by NN prediction of the structural behaviour up to collapse. For every MCS that is required in order to perform the probabilistic constraints check, an NN is trained utilizing available information generated from selected conventional elasto-plastic analyses. The limit state analysis data are processed to obtain input and output pairs, which are used for training the NN. The trained NN is then used to predict the critical load factor due to different sets of basic random variables.

The use of NN is motivated by the approximate concepts inherent in reliability analysis and the time consuming repeated analyses required for MCS. It appears that the use of a properly selected and trained NN can eliminate any limitation on the sample size used for MCS and on the dimensionality of the problem due to the drastic reduction of the computing time achieved.

2. Time invariant structural reliability analysis

The inherent probabilistic nature of design parameters, material properties and loading conditions involved in structural analysis is an important factor that influences structural safety. Reliability analysis leads to safety measures that a design engineer has to take into account due to the aforementioned uncertainties. A time invariant reliability analysis produces the following relationship

$$p_{\mathrm{f}} = p[R < S] = \int_{-\infty}^{\infty} F_{\mathrm{R}}(y) f_{\mathrm{S}}(y)\mathrm{d}y = 1 - \int_{-\infty}^{\infty} F_{\mathrm{S}}(y) f_{\mathrm{R}}(y)\mathrm{d}y, \tag{1}$$

where R denotes the structure's bearing capacity and S the external loads. The randomness of R and S can be described by known probability density functions $f_{\mathrm{R}}(t)$ and $f_{\mathrm{S}}(y)$, respectively, with $F_{\mathrm{R}}(y) = p[R < y]$, $F_{\mathrm{S}}(y) = p[S < y]$ being the cumulative probability density functions of R and S, respectively.

Most often a limit state function is defined as $G(R,S) = S - R$ and the probability of structural failure is given by

$$p_{\mathrm{f}} = p[G(R,S) \geqslant 0] = \int_{G \geqslant 0} f_{\mathrm{R}}(R) f_{\mathrm{S}}(S)\mathrm{d}R\,\mathrm{d}S. \tag{2}$$

It is practically impossible to evaluate R analytically for complex and/or large-scale structures. In such cases the integral of Eq. (2) can be calculated only approximately using either simulation methods, such as the MCS, or approximation methods. First and second order approximation methods (FORM and SORM)

lead to formulations that require prior knowledge of the means and variances of the random variables and the definition of a differentiable failure function. On the other hand, MCS methods require that the probability density functions of all random variables must be known prior to the reliability analysis. For small-scale problems FORM and SORM implementations have been proved very efficient [13], but when the number of random variables increases and the problems become more complex MCS based methods have been proven more reliable [14].

The reliability analysis, employed in this study, is connected to a structural failure criterion of space frames. The failure criterion is considered to be the formation of a mechanism without considering instability effects on the members of the structure. The adopted incremental non-holonomic first order step-by-step limit analysis is based on the generalized plastic node concept. The non-linear yield surface is approximated by a multi-faceted surface [14].

2.1. Monte Carlo simulation

In reliability analysis of structures the MCS method is particularly applicable when an analytical solution is not attainable and the failure domain cannot be expressed or approximated by an analytical form. This is mainly the case in problems of complex nature with a large number of basic variables where all other reliability analysis methods are not applicable. Despite the fact that the mathematical formulation of the MCS is relatively simple and the method has the capability of handling practically every possible case regardless of its complexity, this approach has not received an overwhelming acceptance due to the excessive computational effort that is required. Several sampling techniques, also called variance reduction techniques, have been developed in order to improve the computational efficiency of the method by reducing the statistical error that is inherent in MCS methods and keeping the sample size to the minimum possible. Furthermore, advanced solution methods and parallel processing have been recently implemented having a beneficial effect on the efficiency of MCS [3].

MCS can be stated as follows in structural reliability analysis problems. Expressing the limit state function as $G(x) < 0$, where $x = (x_1, x_2, \ldots, x_M)$ is the vector of the random variables, Eq. (2) can be written as

$$p_{\mathrm{f}} = \int_{G(x) \geqslant 0} f_x(x)\mathrm{d}x, \tag{3}$$

where $f_x(x)$ denotes the joint probability of failure for all random variables. Since MCS is based on the theory of large numbers (N_∞) an unbiased estimator of the probability of failure is given by

$$p_{\mathrm{f}} = \frac{1}{N_\infty} \sum_{j=1}^{N_\infty} I(x_j) \tag{4}$$

in which $I(x_j)$ is an indicator for successful and unsuccessful simulations defined as

$$I(x_j) = \begin{cases} 1 & \text{if } G(x_j) \geqslant 0, \\ 0 & \text{if } G(x_j) < 0. \end{cases} \tag{5}$$

In order to estimate p_{f} an adequate number of N independent random samples is produced using a specific, usually uniform, probability density function of the vector x. The value of the failure function is computed for each random sample x_j and the Monte Carlo estimation of p_{f} is given in terms of sample mean by

$$p_{\mathrm{f}} \cong \frac{N_{\mathrm{H}}}{N}, \tag{6}$$

where N_{H} is the number of successful simulations and N the total number of simulations.

2.2. Importance sampling

Various reduction techniques have been proposed in order to improve the efficiency and the accuracy of the MCS method. Importance sampling (IS) is generally recognized as the most efficient reduction technique [2,5,15]. The key-idea of this technique is to obtain a non-negative sampling density located in the neighbourhood of the most probable failure point. The selection of an appropriate important sampling density function $g_x(x)$ is of critical importance for both the efficiency and the accuracy of the MCS. A successful choice of $g_x(x)$ yields reliable results and reduces significantly the number of simulations, while an inappropriate choice produces inaccurate results. The key-idea of this technique is to obtain a non-negative sampling density located in the neighbourhood of the most probable failure point. Using MCS-IS Eq. (3) can be expressed as

$$p_{\mathrm{f}} = \frac{1}{N}\sum_{j=1}^{N} I(x_j)\frac{f_x(x_j)}{g_x(x_j)}, \tag{7}$$

where $g_x(x)$ is the IS function.

3. Neural networks

Only the basic ideas of NN will be discussed in this study. A more detailed introduction to NN may be found in [16]. Neural net models of learning and the accumulation of expertise have found their way into practical applications in many areas. It appears that a number of computational structures technology applications, that are heavily dependent on extensive computer resources, have been investigated, showing the range of application of neural network capabilities [17–22]. Reliability analysis of ultimate elastic plastic structural response using MCS is a highly intensive computational problem which makes conventional approaches incapable of treating real scale problems even in today's powerful computers. In the present study the use of NN was motivated by the approximation concepts inherent in reliability analysis. The idea here is to train a NN to provide computationally inexpensive estimates of the deterministic and probabilistic constraints check or the limit-elasto-plastic analysis outputs required for the reliability analysis problem. The major advantage of a trained NN over the conventional numerical process, under the provision that the predicted results fall within acceptable tolerances, is that results can be produced in a few clock cycles, requiring orders of magnitude less computational effort than the conventional computational process.

3.1. Back propagation learning algorithm

The basic model for an artificial neuron is shown in Fig. 1. A neural network consists of multiple artificial neurons linked together. In a back propagation (BP) algorithm, learning is carried out when a set of input training patterns is propagated through a network consisting of an input layer, one or more hidden layers and an output layer as shown in Fig. 2 in a fully connected NN. Each layer has its corresponding neurons or nodes and weight connections. A single training pattern is an I/O vector of pairs of input–output values in the entire matrix of I/O training set.

The inputs x_i, $i = 1, 2, \ldots, n$ which are received by the input layer are analogous to the electrochemical signals received by neurons in human brain. In the simplest model these input signals are multiplied by connection weights $w_{p,ij}$ and the effective input $\mathrm{net}_{p,j}$ to neurons is the weighted sum of the inputs

$$\mathrm{net}_{p,j} = \sum_{i=1}^{n} w_{p,ij}\mathrm{net}_{q,i}, \tag{8}$$

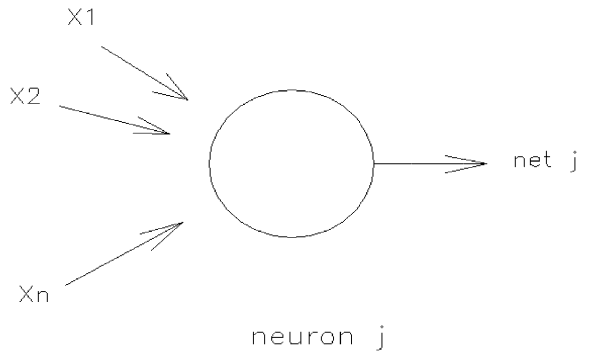

Fig. 1. Basic model for an artificial neuron.

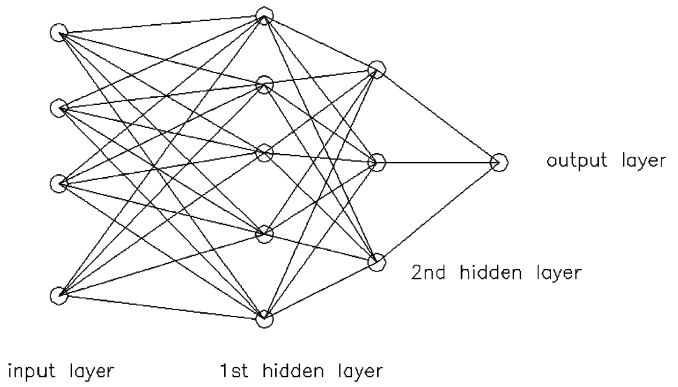

Fig. 2. Three layered fully connected NN configuration.

where $w_{p,ij}$ is the connecting weight of the layer p from the i neuron in the q (source) layer to the j neuron in the p (target) layer, $\text{net}_{q,i}$ is the output produced at the i neuron of the layer q and $\text{net}_{p,j}$ is the output produced at the j neuron in the layer p, as shown in Fig. 3. Inputs x_i correspond to $\text{net}_{q,i}$ for the input layer.

In the biological system, a typical neuron may only produce an output signal if the incoming signal builds up to a certain level. This output is expressed in NN by

$$\text{out}_{p,j} = F(\text{net}_{p,j}), \tag{9}$$

where F is an activation function which produce the output at the j neuron in the p layer. The type of activation function that has been used, for the case of the hidden layers, in the present study is the sigmoid function, while for the case of the output layer the hard limit transfer function is also employed. The sigmoid activation function is given by the expression

$$F(\text{net}_{p,j}) = \frac{1}{1 + \text{e}^{-(\text{net}_{p,j} + b_{p,j})}}, \tag{10}$$

where $b_{p,j}$ is a bias parameter used to modulate the neuron output. The principal advantage of the sigmoid function is its ability to handle both large and small input signals. The determination of the proper weight coefficients and bias parameters is embodied in the network learning process. The weight and bias parameters of the nodes are initialized arbitrarily. The bias parameters are the weights of special connections to each neuron having unity as input value.

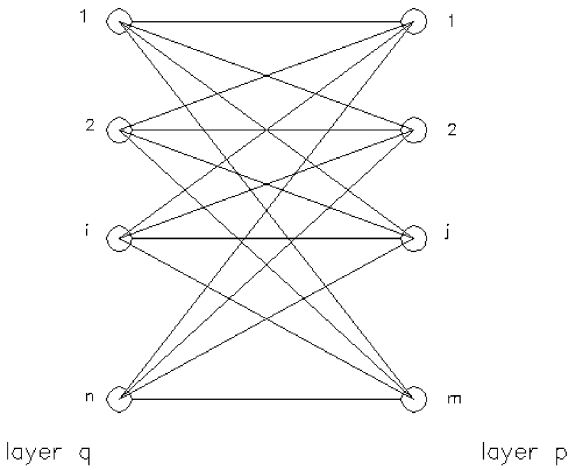

Fig. 3. Connection pattern between two layers.

At the output layer the computed output(s), otherwise known as the observed output(s), are subtracted from the desired or target output(s) to give the error signal

$$\mathrm{err}_{k,i} = \mathrm{tar}_{k,i} - \mathrm{out}_{k,i}, \tag{11}$$

where $\mathrm{tar}_{k,i}$ and $\mathrm{out}_{k,i}$ are the target and the observed output(s) for the node i in the output layer k, respectively. This is called supervised learning. For the output layer the error signal, as given by Eq. (11), is multiplied by the derivative of the activation function, for the neuron in question, to obtain

$$\delta_{k,i} = \mathrm{d}F(\mathrm{net}_{k,i}) \cdot \mathrm{err}_{k,i}, \tag{12}$$

while the derivative of the sigmoid function $\mathrm{d}F$ is given by

$$\mathrm{d}F(\mathrm{net}_{k,i}) = \mathrm{out}_{k,i} \cdot (1 - \mathrm{out}_{k,i}). \tag{13}$$

Subsequently $\delta_{k,i}$ is used for the evaluation of the weight changes in the output layer k according to

$$\Delta w_{k,ji} = \eta \cdot \delta_{k,i} \cdot \mathrm{out}_{p,j}, \tag{14}$$

where η denotes a learning rate coefficient usually selected between 0.01 and 0.9 and $\mathrm{out}_{p,j}$ is the output of node j of the layer p immediately before the output layer. This learning rate coefficient is analogous to the step size parameter in the numerical optimization algorithms.

The changes in the weights may alternatively be expressed according to [16] by

$$\Delta w_{k,ji}^{t+1} = \eta \cdot \delta_{k,i} \cdot \mathrm{out}_{p,j} + \alpha \cdot \Delta w_{k,ji}^{t}, \tag{15}$$

which is adopted in this study, where the superscript t denotes the cycle of the weight modification and α is the momentum term which controls the influence of the previous weight change. For the hidden layers the corresponding weight changes are given by

$$\delta_{q,j} = \mathrm{d}F(\mathrm{net}_{q,j}) \cdot \left(\sum_{i=1}^{n} \delta_{p,i} \cdot w_{p,ji} \right), \tag{16}$$

$$\Delta w_{q,lj}^{t+1} = \eta \cdot \delta_{q,j} \cdot \mathrm{out}_{r,l} + \alpha \cdot \Delta w_{q,lj}^{t}, \tag{17}$$

where $\text{out}_{r,l}$ denotes the output of the neuron l in the hidden layer r, $\Delta w^t_{q,lj}$ is the weight, changes between neuron l in the hidden layer r to neuron j in the hidden layer q which is located between the r and p hidden layers.

After the evaluation of the weight changes the updated values of the weights given by $w^{t+1}_{q,ij} = w^t_{q,ij} + \Delta w^{t+1}_{q,ij}$, are used for the next training cycle until the desired level of error is obtained. The procedure used in this study is the single pattern training where all the weights are updated before the next training pattern (training example) is processed.

3.2. The NN training

In our implementation the main objective is to investigate the ability of the NN either to perform the deterministic and probabilistic constraints check or to predict the structural collapse loads. These objectives comprise the following tasks: (i) Select the proper training set. (ii) Find suitable network architecture. (iii) Determine the appropriate values of characteristic parameters, such as the learning rate and momentum term. For the BP algorithm to provide good results the training set must include data over the entire range of the output space. The appropriate selection of I/O training data is one of the important factors in NN training. Although the number of training patterns may not be the only concern, the distribution of samples is of greater importance. The selection of the I/O training pairs is based on the requirement that the full range of possible results should be represented in the training procedure [23]. In the present study the sample space for each random variable is divided into equally spaced distances for the application of the NN simulation and for the selection of the suitable training pairs.

The number of neurons to be used in the hidden layers is not known in advance and usually is estimated by trial and error. At the first phase of learning it is convenient to start with an increased number of hidden units and then, after achieving the desired convergence, to try to remove some of them in order to find the minimal size of the network which performs the desired task [24].

The learning rate coefficient and the momentum term are two user-defined BP parameters that affect the learning procedure of NN. The training is sensitive to the choice of these net parameters. The learning rate coefficient, employed during the adjustment of weights, is used to speed-up or slow-down the learning process. A bigger learning coefficient increases the weight changes, hence large steps are taken towards the global minimum of error level, while smaller learning coefficients increase the number of steps taken to reach the desired error level. If an error curve shows a downward trend but with poor convergence rate the learning rate coefficient is likely to be too high. Although these learning rate coefficients are usually taken to be constant for the whole net, local learning rate coefficients for each individual layer or unit may be applied as well.

The basic NN configuration employed in this study is selected to have one hidden layer. Tests performed for more than one hidden layer showed no significant improvement in the obtained results. The convergence of the training process is controlled by the prediction error. This is done either with a direct comparison of the predicted with the target results computed by the conventional procedure, also called "exact", or by means of the root mean square (RMS) error which is given by

$$e_{\text{RMS}} = \sqrt{\frac{1}{N_{\text{P}} N_{\text{out}}} \sum_{N_{\text{P}}} \sum_{i=1}^{N_{\text{out}}} (\text{tar}_i - \text{out}_i)^2}, \tag{18}$$

where N_{P} is the total number of I/O pairs in the training set and N_{out} is the number of output units. e_{RMS} gives a measure of the difference between predicted at each NN cycle and "exact" values.

After the selection of the suitable NN architecture and the performance of the training procedure, the network is then used to produce predictions of the deterministic and probabilistic constraints check or the

critical load factor corresponding to different values of the input random variables. The results are then processed by means of MCS to calculate the probability of failure p_f.

4. Reliability-based structural optimization

In deterministic sizing optimization problems the aim is to minimize the weight of the structure under certain deterministic behavioural constraints usually on stresses and displacements. In reliability-based optimal design additional probabilistic constraints are imposed in order to take into account various random parameters. Probabilistic constraints define the feasible region of the design space by restricting the probability that a deterministic constraint is violated within the allowable probability of violation. The probabilistic constraint that is employed in this study enforces the condition that the probability failure of the system is smaller than a certain specified value.

In the present study the reliability-based sizing optimization of large-scale multi-storey 3-D frames is investigated. Thus the overall probability of failure of the structure, as a result of a limit elasto-plastic analysis, is taken as the global reliability constraint. The probabilistic design variables are chosen to be the cross-sectional dimensions of structural members and the material properties modules of elasticity E and yield stress σ_y. Due to engineering practice demands the members are divided into groups having the same design variables. This linking of elements results in a trade-off between the use of more material and the need of symmetry and uniformity of structures due to practical considerations. Furthermore, it has to be taken into account that due to manufacturing limitations the design variables are not continuous but discrete since cross-sections belong to a certain set provided by the manufacturers.

A discrete RBO problem can be formulated in the following form

$$\begin{array}{ll} \min & F(s) \\ \text{subject to} & g_j(s) \leqslant 0, \quad j = 1, \ldots, m, \\ & s_i \in R^d, \quad i = 1, \ldots, n, \\ & p_f \leqslant p_a, \end{array} \tag{19}$$

$F(s)$ is the objective function, s is the vector of geometric design variables, which can take values only from the given discrete set R^d, $g_j(s)$ are the deterministic constraints and p_f is the probability of failure of the structure required to remain below a threshold value (p_a) which comprise the probabilistic constraint. Most frequently the deterministic constraints of the structure are the member stresses and nodal displacements or the inter-storey drifts.

The design variables are selected from the W-shape database comprising of 311 elements. For rigid frames with W-shape cross-sections, the stress constraints, under allowable stress design requirements specified by Eurocode 3 [25], are expressed by the non-dimensional ratio q of the following formulas:

$$q = \frac{f_a}{F_a} + \frac{f_b^y}{F_b^y} + \frac{f_b^z}{F_b^z} \leqslant 1.0 \quad \text{if } \frac{f_a}{F_a} \leqslant 0.15, \tag{20}$$

$$q = \frac{f_a}{F_a} + \frac{C_m f_b^y}{(1 - f_a/F'_e)F_b} + \frac{C_m f_b^z}{(1 - f_a/F'_e)F_b} \leqslant 1.0 \quad \text{if } \frac{f_a}{F_a} > 0.15, \tag{21}$$

where f_a is the computed compressive axial stress, f_b^y, f_b^z are the computed bending stresses for y and z axis, respectively. F'_e is the Euler stress divided by the safety factor 1.10, C_m is a coefficient depending upon element's curvature caused by the applied moments, $F_a = 0.60 \times \sigma_y$ is the allowable axial stress, $F_b = 0.66 \times \sigma_y$ is the allowable bending stress and σ_y is the yield stress. The allowable inter-storey drift is limited to 1.5% of the height of each storey.

The proposed reliability-based sizing optimization methodology proceeds with the following steps:

1. At the outset of the optimization procedure the geometry, the boundaries and the reference loads of the structure under investigation are defined.
2. The constraints are defined in order to formulate the optimization problem as in Eq. (19).
3. The optimization phase is carried out with ES where feasible designs are produced at each generation. The feasibility of the designs is checked for each design vector with respect to both deterministic and probabilistic constraints of the problem.
4. The satisfaction of the deterministic constraints is monitored through a finite element analysis of the structure.
5. The satisfaction of the probabilistic constraints is realized with the reliability analysis of the structure using the MCS technique in order to evaluate its probability of failure.
6. If the convergence criteria for the optimization algorithm are satisfied then the optimum solution has been found and the process is terminated, else the whole process is repeated from step 3 with a new generation of design vectors.

In this work the reliability constraint is related to the ultimate load-carrying capacity of space frame structures. This failure criterion is considered to be the formation of a mechanism as a result of a limit elasto-plastic analysis of the structure without considering member instability effects. The adopted incremental non-holonomic first order step-by-step limit analysis is based on the generalized plastic node concept [26,27]. The non-linear yield surface is approximated by a multi-faceted surface thus avoiding iterations at each load step. In order to prevent the occurrence of very small load steps a second internal and homothetic to the initial yield surface is implemented which form a plastic zone for the activation of the plastic nodes [14].

5. Evolution strategies

The two most widely used optimization algorithms belonging to the class of evolutionary computation that imitate nature by using biological methodologies are the genetic algorithms (GA) and evolution strategies (ES). In this work ES are used as the optimization tool for addressing large-scale RBO problems. ES were introduced in the seventies by Schwefel [28] and have three characteristics that make them differ from other conventional optimization algorithms: (i) in place of the usual deterministic operators, they use randomised operators: mutation, selection, recombination; (ii) instead of a single design point, they work simultaneously with a population of design points; (iii) they can easily handle continuous, discrete and mixed optimization problems [28]. The second characteristic allows for a natural implementation of ES on parallel computing environment.

5.1. ES for discrete optimization problems

In engineering practice the design variables are not continuous because the structural elements are usually manufactured with certain variation of their dimensions. Thus, design variables can only take values from a predefined discrete set. For the solution of this type of problems Thierauf and Cai [29,30] have proposed a modified ES algorithm able to treat discrete optimization problems.

In the multi-membered ES a population of μ parent design vectors will produce λ offsprings through a reproduction phase which involves recombination and mutation. Then the new population of μ parent vectors will be selected either from a temporary population of the current $\mu + \lambda$ individuals ($\mu + \lambda$ selection scheme), or from the set of λ offsprings only (μ, λ selection scheme). In most cases the ES-($\mu + \lambda$) selection

scheme is more efficient than the second type, where the life of each individual is limited to one generation. The ES-(μ, λ) selection scheme is likely to perform better in problems with dynamically varying objective function or in problems where the objective function is noisy.

The basic differences between discrete and continuous ES are focused on the mutation and the recombination operators. The mutation operator ensures that each parent $s_{\mathrm{p}}^{(g)}$, of the current generation g, produces an offspring $s_{\mathrm{o}}^{(g)}$, whose genotype is slightly different from that of the parent

$$s_{\mathrm{o}}^{(g)} = s_{\mathrm{p}}^{(g)} + z^{(g)}, \tag{22}$$

where $z^{(g)} = [z_1^{(g)}, z_2^{(g)}, \ldots, z_n^{(g)}]^{\mathrm{T}}$ is a random vector. The mutation operator in the continuous version of ES produces a normally distributed random change vector $\mathrm{z}^{(g)}$. Each component of this vector has small standard deviation value σ_i and zero mean value. As a result of this assumption there is a possibility that all components of a parent vector may change, but usually these changes are marginal. In the discrete version of ES the random vector $z^{(g)}$ is properly generated in order to force the offspring vector to move to a nearby set of discrete values.

The fact that in discrete optimization problems the difference between any two adjacent values can be relatively large is against the requirement that the variance should be small. For this reason it is suggested that not all the components of a parent vector, but only a few of them (say ℓ) should be randomly changed in every generation. This means that $n - \ell$ components of the randomly changed vector $z^{(g)}$ will have zero value. In other words, the terms of vector $z^{(g)}$ are derived from

$$z_i^{(g)} = \begin{cases} (\kappa + 1)\delta s_i & \text{for } \ell \text{ randomly chosen components,} \\ 0 & \text{for the } n - \ell \text{ other components,} \end{cases} \tag{23}$$

where δs_i is the difference between two adjacent values in the discrete set and κ is a random integer number that follows the Poisson distribution

$$p(\kappa) = \frac{(\gamma)^{\kappa}}{\kappa!}\mathrm{e}^{-\gamma}, \tag{24}$$

γ is the standard deviation as well as the mean value of the random number κ.

In the case of discrete optimization problems one of the following recombination operator schemes are employed

$$\tilde{s}_i = \begin{cases} s_{\mathrm{a},i} \quad \text{or} \quad s_{\mathrm{b},i} \text{ randomly} & (\mathrm{A}), \\ s_{\mathrm{m},i} \quad \text{or} \quad s_{\mathrm{b},i} \text{ randomly} & (\mathrm{B}), \\ s_{\mathrm{b}j,i} & (\mathrm{C}), \\ s_{\mathrm{a},i} \quad \text{or} \quad s_{\mathrm{b}j,i} \text{ randomly} & (\mathrm{D}), \\ s_{\mathrm{m},i} \quad \text{or} \quad s_{\mathrm{b}j,i} \text{ randomly} & (\mathrm{E}), \end{cases} \tag{25}$$

$\tilde{s}_i$ is the ith component of the temporary parent vector $\tilde{s}$, $s_{\alpha,i}$ and $s_{\mathrm{b},i}$ are the ith components of two parent vectors s_{a} and s_{b} randomly chosen from the population. The vector s_{m} is not randomly chosen but is the best of the μ parent vectors in the current generation. In case of the recombination scheme C, $\tilde{s}_i = s_{\mathrm{b}j,i}$ means that the ith component of $\tilde{s}$ is chosen randomly from the ith components of all μ parent vectors. From the temporary parent $\tilde{s}$ an offspring can then be created following the mutation operator.

The discrete optimization procedure terminates when one of the following heuristic criteria is satisfied [29]: (i) when the best value of the objective function in the last $4n\mu/\lambda$ generations remains unchanged, (ii) when the mean value of the objective values from all parent vectors in the last $2n\mu/\lambda$ generations has not been improved by less than a given value ε_{b}, (iii) when the relative difference between the best objective function value and the mean value of the objective function values from all parent vectors in the current generation is less than a given value ε_{c}, (iv) when the ratio μ_{b}/μ has reached a given value ε_{d} where μ_{b} is the

number of the parent vectors in the current generation with the best objective function value. In the present study ε_b and ε_c are taken 0.0001 and ε_d takes the values from 0.5 to 0.8.

5.2. ES in structural optimization problems

In structural optimization problems, where the objective function and the constraints are particularly highly non-linear functions of the design variables, the computational effort spent in gradient calculations needed for the mathematical programming algorithms is usually high. In a recent study by Papadrakakis et al. [12] it was found that probabilistic search methods in structural optimization are computationally efficient even if large number of optimization steps is needed to reach the optimum. These optimization steps are computationally less expensive than those of mathematical programming algorithms since they do not need gradient information. This property of probabilistic search methods is of greater importance in the case of RBO problems since the calculation of the derivatives of the probabilistic constraints is extremely time-consuming. Furthermore, probabilistic methodologies are more capable of finding the global optimum due to their random search, whereas mathematical programming algorithms may be trapped in local optima.

The ES optimization procedure starts with a set of parent vectors and if any of these parent vectors gives an infeasible design then it is modified until it becomes feasible. Subsequently, the offspring design vectors are generated and checked if they are in the feasible region. According to $(\mu + \lambda)$ selection scheme the values of the objective function of the parent and the offspring vectors in every generation are compared and the worst vectors are rejected, while the remaining ones are considered to be the parent vectors of the new generation. This procedure is repeated until the chosen termination criterion is satisfied. The ES algorithm for structural optimization applications can be stated as follows:

1. *Selection step:* selection of s_i $(i = 1, 2, \ldots, \mu)$ parent design vectors.
2. *Analysis step:* solve $K(s_i)x_i = b$ $(i = 1, 2, \ldots, \mu)$.
3. *Constraints check:* if satisfied continue, else change s_j and go to *step 1*.
4. *Offspring generation:* generate s_j $(j = 1, 2, \ldots, \lambda)$ offspring design vectors.
5. *Analysis step:* solve $K(s_j)x_j = b$ $(j = 1, 2, \ldots, \lambda)$.
6. *Constraints check:* if satisfied continue, else change s_j and go to *step 4*.
7. *Selection step:* selection of the next generation parent design vectors.
8. *Convergence check:* if satisfied stop, else go to *step 4*.

An important characteristic of ES is that instead of a single design point, like most of the conventional optimization algorithms, they work simultaneously with a population of design points. This allows the natural implementation of the ES optimization procedure in parallel computing environments where the finite element analyses of the structure corresponding to the same population are performed independently and concurrently. The most straightforward parallel implementation of ES is to assign each individual of the current population to a processor without any need of inter-processor communication during the analysis phase.

In a distributed memory computing environment the natural parallel implementation of ES can be realized provided that each processor's memory capacity is adequate to accommodate the matrices and vectors required by the solution algorithm. In a shared memory environment, on the other hand, the number of processors employed is related to the storage limitations since the total memory required is the corresponding memory required for serial computations multiplied by the number of processors p. In the present study the parallel computations were performed on a Silicon Graphics Power Challenge shared memory computer where the number of processors activated is equal to the number of the parent or offspring design vectors since $\mu = \lambda$.

5.3. Reliability-based structural optimization using MCS, ES and NN

In reliability analysis of elasto-plastic structures using MCS the computed critical load factors are compared to the corresponding external loading leading to the computation of the probability of structural failure. The probabilistic constraints enforce the condition that the probability of a local failure of the system or the global system failure is smaller than a certain value (i.e. 10^{-5}–10^{-3}). In this work the overall probability of failure of the structure, as a result of limit elasto-plastic analyses, is taken as the global reliability constraint. The probabilistic design variables are chosen to be the cross-sectional dimensions of the structural members and the material properties (E, σ_y).

MCS requires a number of limit elasto-plastic analyses that can be dealt independently and concurrently. This allows the natural implementation of the MCS method in parallel computing environment as well. The most straightforward parallel implementation of the MCS method is to assign one limit elasto-plastic analyses to a processor without any need of inter-processor communication during the analysis phase.

5.3.1. NN used for deterministic and probabilistic constraints check

In this methodology, a trained NN utilizing information generated from a number of properly selected design vectors is used to perform both the deterministic and probabilistic constraints checks during the optimization process. After the selection of the suitable NN architecture the training procedure is performed using a number (M) of data sets, in order to obtain the I/O pairs needed for the NN training. The trained NN is then applied to predict the response of the structure in terms of deterministic and probabilistic constraints checks due to different sets of design variables.

The combined ES–NN optimization procedure is performed in two phases. The first phase includes the training set selection, the corresponding structural analysis and MCS for each training set required to obtain the necessary I/O data for the NN training, and finally the training and testing of a suitable NN configuration. The second phase is the ES optimization stage where the trained NN is used to predict the response of the structure in terms of the deterministic and probabilistic constraints checks due to different sets of design variables.

This ES–NN methodology can be described with the following algorithm 1:

- NN training phase:
  1. *Training set selection step:* select M input patterns.
  2. *Deterministic constraints check:* perform the check for each input pattern vector.
  3. *MCS step:* perform MCS for each input pattern vector.
  4. *Probabilistic constraints check:* perform the check for each input pattern vector.
  5. *Training step:* training of the NN.
  6. *Testing step:* test the trained NN.
- ES–NN optimization phase:
  1. *Parents initialization.*
  2. *NN (deterministic–probabilistic) constraints check:* all parents become feasible.
  3. *Offspring generation.*
  4. *NN (deterministic–probabilistic) constraints check:* if satisfied continue, else go to *step 3*.
  5. *Parents' selection step.*
  6. *Convergence check.*

5.3.2. NN prediction of the critical load in structural failure

In the second methodology the limit elasto-plastic analyses required during the MCS are now replaced by NN prediction of the structural behaviour up to collapse. For every MCS an NN is trained utilizing available information generated from selected conventional elasto-plastic analyses. The limit state analysis

data is processed to obtain input and output pairs, which are used for training the NN. The trained NN is then used to predict the critical load factor due to different sets of basic random variables.

At each ES cycle (generation) a number of MCS are carried out. In order to replace the time consuming limit elasto-plastic analyses by predicted results obtained with a trained NN, a training procedure is performed based on the data collected from a number of conventional limit elasto-plastic analyses. After the training phase is concluded the trained NN predictions replace the conventional limit elasto-plastic analyses, for the current design. For the selection of the suitable training pairs, the sample space for each random variable is divided into equally spaced distances. The central points within the intervals are used as inputs for the limit state analyses.

This ES–NN methodology can be described with the following algorithm 2:

1. *Parents initialization*.
2. *Deterministic constraints check:* all parents become feasible.
3. *MCS step:*
   3a. *Selection of the NN training set*.
   3b. *NN training for the limit load*.
   3c. *NN testing*.
   3d. *Perform MCS using NN*.
4. *Probabilistic constraints check:* all parents become feasible.
5. *Offspring generation*.
6. *Deterministic constraints check:* if satisfied continue, else go to *step 5*.
7. *MCS step:*
   7a. *Selection of the NN training set*.
   7b. *NN training for the limit load*.
   7c. *NN testing*.
   7d. *Perform MCS using NN*.
8. *Probabilistic constraints check:* if satisfied continue, else go to *step 5*.
9. *Parents' selection step*.
10. *Convergence check*.

6. Numerical results

One characteristic 3-D building frame has been tested in order to illustrate the efficiency of the proposed methodologies for reliability-based sizing optimization problems. The cross-section of each member of the space frame considered is assumed to be a W-shape and for each structural member one design variable is allocated corresponding to a member of the W-shape data base. The objective function is the weight of the structure. The deterministic constraints are imposed on the inter-storey drifts and, for each group of structural members, on the maximum non-dimensional ratio q of Eqs. (20) and (21) which combines axial forces and bending moments. The values of allowable axial and bending stresses are $F_a = 150$ MPa and $F_b = 165$ MPa, respectively, whereas the allowable inter-storey drift is restricted to 1.5% of the height of each storey.

The probabilistic constraint is imposed on the probability of structural collapse due to successive formation of plastic nodes and is set to $p_a = 0.001$. The probability of failure caused by uncertainties related to material properties, geometry and loads of the structures is estimated using MCS with the IS technique. External loads, yield stresses, elastic moduli and the dimensions of the cross-sections of the structural members are considered to be random variables. The loads follow a log-normal probability density function, while random variables associated with material properties and cross-section dimensions follow a

normal probability density function. The required IS function $g_x(x)$ for the loads is assumed to follow a normal distribution.

In the tables showing the results of the test example, DBO stands for the conventional deterministic optimization approach, RBO stands for the conventional reliability-based optimization approach, while RBO-NNi corresponds to the proposed reliability-based optimization with NN incorporating algorithm i $(i = 1, 2)$.

6.1. Twenty-storey space frame

The twenty-storey space frame shown in Fig. 4 consists of 1020 members with 2400 degrees of freedom. This example is selected in order to show the efficiency of the proposed methodologies in relatively large-scale RBO problems. The basic load of the structure is a uniform vertical load of 4.78 kPa at each storey and a horizontal pressure of 0.956 kPa acting on the x–z face of the frame. The members of the frame are divided into 11 groups, as shown in Fig. 4, and the total number of design variables is 11. The deterministic constraints are 23, two for the stresses of each element group and one for the inter-storey drift. The type of probability density functions, mean values, and variances of the random parameters are shown in Table 1. A typical load–displacement curve of a node in the top-floor is depicted in Fig. 5, corresponding to the following design variables: 14WF176, 14WF158, 14WF142, 14WF127, 12WF106, 12WF85, 10WF60, 8WF31, 12WF27, 16WF36, 16WF36.

For this test case the $(\mu + \lambda)$-ES approach is used with $\mu = \lambda = 10$, while a sample size of 500 and 1000 simulations is taken for the MCS with the important sampling technique. Table 2 depicts the performance of the optimization procedure for this test case. As can be seen the probability of failure corresponding to the optimum computed by the deterministic optimization procedure is much larger than the specified value

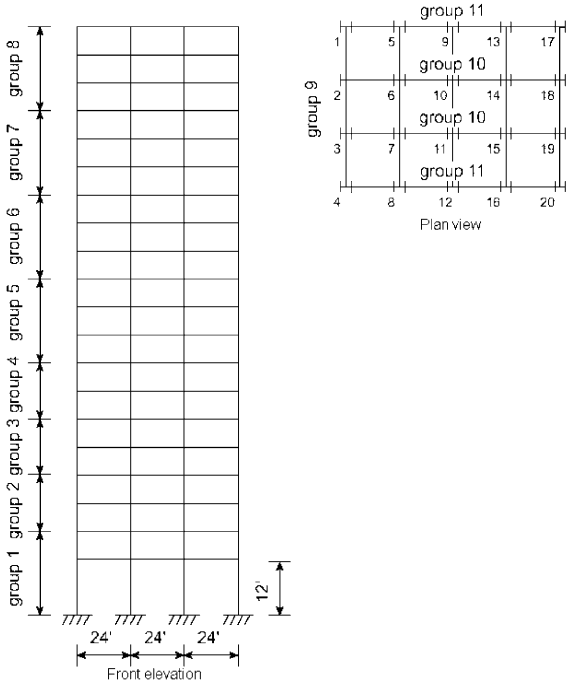

Fig. 4. Description of the twenty-storey frame.

Table 1
Characteristics of the random variables

Random variables	Probability density function (pdf)	Mean value	Standard deviation (σ)
E	N	200	0.10E
σ_y	N	25.0	$0.10\sigma_y$
Design variables	N	s_i	$0.1s_i$
Loads	Log-N	5.2	0.2

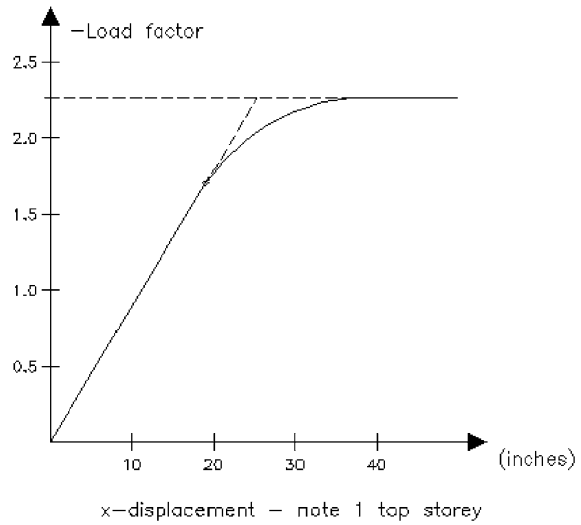

Fig. 5. Load–displacement curve.

Table 2
Performance of the methods

Optimization procedure	ES Gens.	p_f[a]	Optimum weight (kN)	Sequential time (h)	Parallel time (h)		
					$p = 5$	$p = 10$	$p = 20$
DBO	83	0.197×10^{-0}	6771	2.0	0.7	0.3	0.3
RBO (500 siml.)	126	0.103×10^{-2}	9114	141.0	28.4	14.1	7.1
RBO-NN1 (500 siml.)	129	0.102×10^{-2}	9121	34.5	7.2	3.5	1.8
RBO-NN2 (500 siml.)	126	0.103×10^{-2}	9114	15.8	3.3	1.7	0.9
RBO (1000 siml.)	120	0.103×10^{-2}	9156	250.3	50.1	25.1	12.6
RBO-NN1 (1000 siml.)	127	0.101×10^{-2}	9172	68.5	13.8	6.9	3.5
RBO-NN2[b]	122	0.97×10^{-3}	9255	17.0	4.1	2.2	1.2

[a] For 100,000 simulations using the NN2 scheme.
[b] For 100,000 simulations.

of 10^{-3}. For this example the increase on optimum weight achieved, when probabilistic constraints are considered, is approximately 26% of the deterministic one, as it can be observed from Table 2. For the

application of the RBO-NN1 methodology the number of NN input units is equal to the number of design variables. Consequently the NN configuration used in this case has one hidden layer with 15 nodes resulting in an 11-15-1 NN architecture which is used for all runs. The training set consists of 200 training patterns capturing the full range of possible designs.

For the application of the RBO-NN2 methodology the number of NN input units is equal to the number of random variables, whereas one output unit is needed corresponding to the critical load factor. Consequently the NN configuration with one hidden layer results in a 3-7-1 NN architecture which is used for all runs. The number of conventional step-by-step limit analysis calculations performed for the training of NN is 60 corresponding to different groups of random variables properly selected from the random field. As can be seen from Table 2 the proposed RBO-NN2 optimization scheme manages to achieve the optimum weight in one-tenth of the CPU time required by the conventional RBO procedure in sequential computing implementation.

Table 2 also depicts the performance of the proposed methodologies in a straightforward parallel mode, with 5, 10 or 20 processors in which 5, 10 or 20 MCSs are performed independently and concurrently. It can be seen that the parallel versions of RBO, RBO-NN1 and RBO-NN2 reached the perfect speedup irrespective of the number of processors used.

7. Conclusions

In most cases optimum design of structures is based on deterministic parameters and is focused on the satisfaction of the associated deterministic constraints. So far many articles have been devoted to this research field and efficient methods have been presented. When many random factors affect the design, the manufacturing and the life of a structure the deterministic optimum cannot be considered as a realistic optimum design when a number of uncertain parameters have an important influence on its structural behaviour. In order to find a more realistic optimum the designer has to take into account all necessary random parameters and via a reliability analysis of the structure to determine its optimum design taking into account several probabilistic constraints such as the probability of failure of the structure. Only after forming and solving this reliability based optimization problem, even with additional cost in weight and computing time, a more realistic optimum structural design can be found.

The aim of the proposed reliability based optimization procedure was threefold. To reach an optimized design with controlled safety margins with regard to various model uncertainties, while at the same time minimizing the weight of the structure and reducing substantially the required computational effort. The solution of realistic reliability based optimization problems in structural mechanics is an extremely computationally intensive task. In the test example considered in this study the conventional reliability based optimization procedure was found over seventy times more expensive than the corresponding deterministic optimization procedure. The goal of decreasing the computational cost by one order of magnitude in sequential mode was achieved using: (i) NN predictions to perform both deterministic and probabilistic constraints check, or (ii) NN predictions to perform the structural analyses involved in MCS. Furthermore, the achieved reduction in computational time was almost two orders of magnitude in parallel mode with the proposed NN methodologies.

References

[1] G.I. Schueller, Structural reliability—recent advances, Seventh International Conference on Structural Safety and Reliability, ICOSSAR'97, Kyoto, Japan, 1997.

[2] J.E. Hurtado, A.H. Barbat, Simulation methods in stochastic mechanics, in: J. Marczyk (Ed.), Computational Stochastic Mechanics in a Meta-computing Perspective, CIMNE, Barcelona, 1997, pp. 93–116.

[3] Y. Tsompanakis, M. Papadrakakis, Large-scale reliability based structural optimization, Journal of Structural and Multidisciplinary Optimization, in press.
[4] W. Li, L. Yang, An effective optimization procedure based on structural reliability, Comp. Struct. 52 (5) (1994) 1061–1071.
[5] D. Frangopol, Interactive reliability based structural optimization, Comp. Struct 19 (4) (1984) 559–563.
[6] E. Pulido, T.L. Jacobs, E.C. Prates De Lima, Structural reliability using Monte-Carlo simulation with variance reduction techniques on elastic–plastic structures, Comp. Struct. 43 (1992) 419–430.
[7] M. Gasser, G.I. Schueller, Reliability based optimization (RBO) of structures and mechanical components, in: K. Marti, P. Kall (Eds.), Stochastic Programming Methods and Technical Applications, Lecture Notes in Economics and Mathematical Science, vol. 458, Springer, Berlin, 1998, pp. 80–103.
[8] R. Stocki, A. Siemaszko, M. Kleiber, Interactive methodology for reliability-based structural design and optimisation, Shock Comput. Assisted Mech. Engrg. Sci. 6 (1) (1999) 39–62.
[9] C. Thampan, K. Prasad Varma, C.S. Krishnamoorthy, System reliability-based configuration optimization of trusses, J. Struct. Engrg. 127 (8) (2001) 947–958.
[10] S. Hendawi, D.M. Frangopol, Design of composite hybrid plate girder bridges based on reliability and optimization, Struct. Safety 15 (1/2) (1994) 149–165.
[11] M. Papadrakakis, Y. Tsompanakis, E. Hinton, J. Sienz, Advanced solution methods in topology optimization and shape sensitivity analysis, J. Engrg. Comput. 13 (5) (1996) 57–90.
[12] M. Papadrakakis, Y. Tsompanakis, N.D. Lagaros, Structural shape optimization using evolution strategies, Engrg. Optimization J. 31 (1999) 515–540.
[13] D.M. Frangopol, F. Moses, Reliability-based structural optimization, in: H. Adeli (Ed.), Advances in Design Optimization, Chapman-Hall, London, 1994, pp. 492–570.
[14] M. Papadrakakis, V. Papadopoulos, A computationally efficient method for the limit elasto plastic analysis of space frames, Comput. Mech. 16 (2) (1995) 132–141.
[15] C.G. Bucher, Adaptive sampling—an iterative fast Monte Carlo procedure, Struct. Safety 5 (1988) 119–126.
[16] D.E. Rummelhart, J.L. McClelland, Parallel Distributed Processing, Foundations, vol. 1, The MIT Press, Cambridge, 1986.
[17] R.C. Shieh, Massively parallel structural design using stochastic optimization and mixed neural net/finite element analysis methods, Comput. Syst. Engrg. 5 (4–6) (1994) 455–467.
[18] H. Adeli, H.S. Park, Optimization of space structures by neural dynamics, Neural Networks 8 (5) (1995) 769–781.
[19] J.E. Stephens, D. VanLuchene, Integrated assessment of seismic damage in structures, Microcomp. Civil Engrg. 9 (2) (1994) 119–128.
[20] B.H.V. Topping, A. Bahreininejad, Neural Computing for Structural Mechanics, Saxe Coburg, UK, 1997.
[21] P.S. Theocharis, P.D. Panagiotopoulos, Neural networks for computing in fracture mechanics. Methods and prospects of applications, Comput. Meth. Appl. Mech. Engrg. 106 (1993) 213–228.
[22] Z. Waszczyszyn, Some recent and current problems of neurocomputing in civil and structural engineering, in: B.H.V. Topping (Ed.), Advances in Computational Structures Technology, CIVIL-COMP Press, Edinburgh, 1996, pp. 43–58.
[23] M. Papadrakakis, V. Papadopoulos, N.D. Lagaros, Structural reliability analysis of elastic-plastic structures using neural networks and Monte Carlo simulation, Comput. Meth. Appl. Mech. Engrg. 136 (1996) 145–163.
[24] Y. Hirose, K. Yamashita, S. Hijiya, Back-propagation algorithm which varies the number of hidden units, Neural Networks 4 (1991) 61–66.
[25] Eurocode 3, Design of steel structures, Part 1.1: General rules for buildings, CEN, ENV 1993-1-1/1992.
[26] Y. Ueda, T. Yao, The plastic node method: a new method of plastic analysis, Comput. Meth. Appl. Mech. Engrg. 34 (1982) 1089–1104.
[27] J.G. Orbinson, W. McGuire, J.F. Abel, Yield surface applications in non-linear steel frames analysis, Comput. Meth. Appl. Mech. Engrg. 33 (1982) 557–573.
[28] H.P. Schwefel, Numerical Optimization for Computer Models, Wiley, Chichester, UK, 1981.
[29] G. Thierauf, J. Cai, A two level parallel evolution strategy for solving mixed-discrete structural optimization problems, The 21st ASME Design Automation Conference, Boston MA, September 17–21, 1995.
[30] G. Thierauf, J. Cai, Structural optimization based on evolution strategy, in: M. Papadrakakis, G. Bugeda (Eds.), Advanced Computational Methods in Structural Mechanics, CIMNE, Barcelona, 1996, pp. 266–280.