ARTICLE IN PRESS

Available at www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

ELSEVIER

Applied Soft Computing xxx (2003) xxx–xxx

Applied Soft Computing

www.elsevier.com/locate/asoc

Soft computing methodologies for structural optimization

Manolis Papadrakakis*, Nikos D. Lagaros

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

Received 29 May 2002; received in revised form 15 April 2003; accepted 16 April 2003

Abstract

The paper examines the efficiency of soft computing techniques in structural optimization, in particular algorithms based on evolution strategies combined with neural networks, for solving large-scale, continuous or discrete structural optimization problems. The proposed combined algorithms are implemented both in deterministic and reliability based structural optimization problems, in an effort to increase the computational efficiency as well as the robustness of the optimization procedure. The use of neural networks was motivated by the time-consuming repeated finite element analyses required during the optimization process. A trained neural network is used to perform either the deterministic constraints check or, in the case of reliability based optimization, both the deterministic and the probabilistic constraints checks. The suitability of the neural network predictions is investigated in a number of structural optimization problems in order to demonstrate the computational advantages of the proposed methodologies.

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

1. Introduction

Evolutionary algorithms have successfully been applied to a variety of applications in computational mechanics. The effectiveness of evolutionary type algorithms in structural shape optimization has been examined in a number of cases. In shape optimization of truss structures [1], aerodynamic improvement [2,3] and shape optimization of 2D mechanical parts [4,5]. Evolutionary algorithms have also been applied in a number of sizing structural optimization problems [6–8]. However, the relevant literature when probabilistic constraints are taken into consideration is rather limited [9,10]. Optimization of large-scale structures, such as sizing optimization of multi-storey 3-D frames or shape optimization of 2-D mechanical parts, is a computationally intensive task, especially when both deterministic and probabilistic constraint functions are involved.

In the present work the application of evolution strategies (ES) combined with neural networks (NN) is investigated, both in deterministic and reliability based structural optimization problems, in an effort to increase the computational efficiency as well as the robustness of the optimization procedure. The reason for introducing reliability theory into structural engineering and structural optimization is to account in a more rational way of all existing uncertainties that can influence structural response as well as of the fact that the loadings applied to a structure are not known with the desirable degree of precision. Reliability is recognized as a safety constraint in structural engineering and an optimum design should balance both cost and

* Corresponding author. Tel.: +30-1-7721694;
fax: +30-1-7721693.
E-mail addresses: mpapadra@central.ntua.gr (M. Papadrakakis), nlagaros@central.ntua.gr (N.D. Lagaros).

1568-4946/$ – see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/S1568-4946(03)00040-1

safety. In the case of deterministic based design, stress and displacement constraints are considered accepting the design code safety factors without introducing reliability as explicit design constraints, while in the case of the reliability based design additional probabilistic constraints related to the strength of the structure are considered.

The basic idea of the proposed methodologies in this work is to employ a properly trained NN for performing the time consuming calculations that are encountered repeatedly during the optimization process. In the case of deterministic based structural optimization (DBO) the use of NN was motivated by the time-consuming repeated finite element (FE) analyses required for the optimization process. In the case of reliability-based optimization (RBO) both deterministic and probabilistic constraint functions have to be taken into consideration. The use of NN in RBO is motivated by the approximations that are inherent in reliability analysis and the time consuming repeated analyses required by Monte Carlo simulation (MCS). It appears that the use of a properly selected and trained NN can eliminate any limitation on the dimensionality of the FE model and on the sample size required for MCS. Thus, in the case of RBO, the robustness of the optimization algorithm is improved, since it provides with no extra computational cost more accurate calculation of the probability constraints.

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

$$\begin{aligned} p_{\mathrm{f}} &= p[R < S] = \int_{-\infty}^{\infty} F_R(y) f_S(y)\,\mathrm{d}y \\ &= 1 - \int_{-\infty}^{\infty} F_S(y) f_R(y)\,\mathrm{d}y \end{aligned} \tag{1}$$

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 $f_R(t)$ and $f_S(y)$, respectively, with $F_R(y) = p[R < y]$, $F_S(y) = p[S < y]$ 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

$$p_{\mathrm{f}} = p[G(R, S) \geq 0] = \int_{G\geq 0} f_R(R) f_S(S)\,\mathrm{d}R\,\mathrm{d}S \tag{2}$$

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 Monte Carlo Simulation, 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 [11], but when the number of random variables increases and the problems become more complex MCS-based methods have been proven more reliable [12].

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 [12].

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 [10].

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

$$p_f = \int_{G(x)\geq 0} f_x(x)\,\mathrm{d}x \tag{3}$$

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

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

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

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

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

$$p_f \cong \frac{N_H}{N} \tag{6}$$

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 [13–15]. The key-idea of this technique is to obtain a non-negative sampling density located in the neighborhood 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 neighborhood of the most probable failure point. Using MCS-IS Eq. (3) can be expressed as

$$p_f = \frac{1}{N}\sum_{j=1}^{N} I(x_j)\frac{f_x(x_j)}{g_x(x_j)} \tag{7}$$

where $g_x(x)$ is the importance sampling function.

3. Neural networks

A neural network attempts to create a desired relation for an input/output set of m learning patterns. This set which is called training set consists of a finite number of m pairs (inp, tar) $\in R^k \times R^\ell$. The first coordinate is a position in k-dimensional space, corresponding to the input space, and the second coordinate is a position in ℓ-dimensional space, corresponding to the desired or target space. The algorithm that is usually used in order to form the relation $R^k \to R^\ell$ between those two spaces is the back propagation algorithm. This algorithm tries to determine a set of parameters called weights, in order to achieve the right response for each input vector applied to the network. If the training is successful, application of a set of inputs to the network produces the desired set of outputs. Thus, in the case of NN training, w corresponds to the weight matrix defining the parameters to be determined, while the objective function can be defined as follows

$$E(w) = \frac{1}{2m}||E(w)||^2 \tag{8}$$

where the terms of the vector $E(w) = [E_1(w), E_2(w), \ldots, E_m(w)]^T$ have to minimized. E_i denotes the residual between the value of the approximating function and the desired value. This term is determined by the following expression

$$E_i(w) = \sum_{j=1}^{\ell}[\text{out}_j(\text{inp}_i, w) - \text{tar}_{i,j}] \tag{9}$$

where inp_i is a k-dimensional input vector, tar_i the desired response corresponding to the ith input, $\text{tar}_{i,j}$ the desire response of the jth node of the output vector for the ith input pattern, while out is the response of the network for the current values of the weight parameters.

3.1. Minimization algorithms

The numerical minimization algorithms used for the solution of the problem of Eq. (8) generate a sequence of weight parameters through an iterative procedure. To apply an algorithmic operator A we need a starting weight matrix $w^{(0)}$, while the iteration formula can be written as follows

$$w^{(t+1)} = A(w^{(t)}) = w^{(t)} + \Delta w^{(t)} \tag{10}$$

All numerical methods applied for the solution of Eq. (8) are based on the above formula. The changing part of the algorithm $\Delta w^{(t)}$ is further decomposed into two parts as

$$\Delta w^{(t)} = a_t d^{(t)} \tag{11}$$

where $d^{(t)}$ is a desired search direction of the move and a_t the step size in that direction.

The algorithms most frequently used in the NN training are the steepest descent, the conjugate gradient, the Newton and the Levenberg–Marquard methods with the following direction vectors:

Steepest descent method: $d^{(t)} = -\nabla E(w^{(t)})$
Conjugate gradient method: $d^{(t)} = -\nabla E(w^{(t)}) + \beta_{t-1}d^{(t-1)}$ where β_t is defined as follows

$$\beta_{t-1} = \begin{cases} \nabla E_t \cdot (\nabla E_t - \nabla E_{t-1})/d^{(t-1)} \cdot (\nabla E_{t-1} - \nabla E_t) & \text{Hestenes–Stiefel} \\ \nabla E_t \cdot (\nabla E_t - \nabla E_{t-1})/\nabla E_{t-1} \cdot \nabla E_{t-1} & \text{Polak–Ribiere} \\ \nabla E_t \cdot \nabla E_t/\nabla E_{t-1} \cdot \nabla E_{t-1} & \text{Fletcher–Reeves} \end{cases}$$

Newton method: $d^{(t)} = -[H(w^{(t)})]^{-1}\nabla E(w^{(t)})$
Levenberg–Marquard method: $d^{(t)} = -[H(w^{(t)}) + \lambda_t I]^{-1}\nabla E(w^{(t)})$

where λ_t is a positive constant and $\nabla E(w^{(t)})$ is the gradient of the function E.

$$\nabla E(w) = J(w)^{\mathrm{T}} E(w) \tag{12}$$

$H(w)$ is the Hessian matrix of the function E

$$\nabla^2 E(w) = H(w) = J(w)^{\mathrm{T}} J(w) + \sum_{i=1}^{m} E_i(w) H_i(w) \tag{13}$$

$J(w)$ is the Jacobian matrix of vector function $E(w)$ and $H_i(w)$ is the Hessian matrix of the component function $E_i(w)$. In the present study the steepest descent and the Levenberg–Marquard [16] methods are used.

3.2. Improving generalization

One of the problems that occur during neural network training is called over-fitting. The error on the training set is driven to a very small value, but when new data is presented to the network the error is large. The network has memorized the training samples, but it is not capable to generalize to new ones. A method for improving generalization is called regularization. This involves modifying the objective function of the Eq. (8). The typical objective function that is used for training feed-forward neural networks is the mean sum of squares of the network errors. It is possible to improve generalization if we modify the performance function by adding a term that consists of the mean of the sum of squares of the network weights and biases as follows

$$M(w) = \gamma E(w) + (1 - \gamma) E_w(w) \tag{14}$$

where λ is the performance ratio, and

$$E_w(w) = \frac{1}{2}\sum_i w_i^2 \tag{15}$$

Using this modified function will cause the network to have smaller weights and biases, and this will force the network response to be smoother and less likely to over-fit. The problem with regularization is that it is difficult to determine the optimum value for the performance ratio parameter λ. If we make this parameter too large, we may get overfitting. If the ratio is too small, the network will not adequately fit the training data.

One approach to determine the regularization parameters in an automated fashion is the Bayesian framework proposed by David MacKay [17]. In this framework, the weights and biases of the network are assumed to be random variables with specified distributions. The regularization parameters are related to the unknown variances associated with these distributions. We can then estimate these parameters using statistical techniques. A detailed discussion of the use of Bayesian regularization, in combination with Levenberg–Marquardt training that have been employed in the current study, can be found in [18].

3.3. The training set

In our implementation the main objective is to investigate the ability of the NN either to perform the probabilistic and/or deterministic constraints check or to predict the collapse loads of the structures. 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. In an effort to increase the robustness as well as the computational efficiency of the NN procedure, in the case of DBO, two types of selection are used in this study: (i) the training set is chosen automatically based on a uniform distribution of the design variables in the design space, (ii) the training set is chosen automatically based on a Gaussian distribution of the design variables around the midpoints of the design space. In the case of RBO 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.

4. Structural optimization

In deterministic-based optimization problems the aim is to minimize the weight or the volume of the structure under certain deterministic behavioral 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.

4.1. Deterministic-based structural optimization

4.1.1. Shape optimization

Shape optimization attempts to integrate geometrical modeling, structural analysis, and optimization into one complete and automated computer aided design process. It determines the shape of the boundary of a two- or three-dimensional structural component of minimum mass under constraints on geometry and structural responses such as stress, displacements and natural frequencies. A number of researchers in the past have been dealt with the structural shape optimization problem introducing new methods and new types of shape optimization problems [19–23]. The mathematical formulation of a typical structural shape optimization problem with respect to the design variables, the objective and constraint functions can be expressed in standard mathematical terms as a non-linear programming problem as follows:

$$\begin{array}{lll} \text{min} & F(s), & \\ \text{subject to} & h_j(s) \leq 0, & j = 1, \ldots, m \\ & s_i^l \leq s_i \leq s_i^u, & i = 1, \ldots, n \end{array} \qquad (16)$$

where s is the vector of design variables, $F(s)$ the objective function to be minimized (or maximized), $h_j(s)$ the behavioral constraints, s_i^l and s_i^u are the lower and the upper bounds of a typical design variable s_i.

The set of design variables gives a unique definition of a particular design. The selection of design variables is very important in the optimization process. The designer has to decide a priori where to allow design changes and to evaluate how these changes should take place by defining the location of the design variables and the moving directions. The use of the coordinates at key points of the curves that define the shape of the structural model as design variables leads to fewer design variables and more freedom in controlling the shape of the structure.

It is an issue of extreme importance to formulate the optimization problem correctly otherwise unrealistic solutions may be found. Normally, it is necessary to constrain some function of the stresses (e.g. the principal stress) so that it will not exceed a specified value throughout the entire structure. From a practical point of view a finite number of so-called 'stress constraint points' is selected, where the condition is enforceable. These points are either some predefined points within the domain, or some boundary nodes. Other type of constraints, like displacement or frequency constraints, can also be imposed depending on the type of problem. Usually the constraint functions and their derivatives are normalized in order to improve the performance of the optimizer.

The shape optimization methodology proceeds with the following steps: (1) at the outset of the optimization, the geometry of the structure under investigation has to be defined. The boundaries of the structure are modeled using cubic B-splines which, in turn, are defined by a set of key points. Some of the coordinates of these key points will be the design variables which may or may not be independent to each other. (2) An automatic mesh generator is used to create a valid and complete FE model. A finite element analysis, is then carried out and the displacements and stresses are evaluated. In order to increase the accuracy of the analysis an h-type adaptivity analysis may be incorporated in this stage. (3) The design variables are being optimized. If the convergence criteria for the optimization algorithm are satisfied, then the optimum solution has been found and the process is terminated, else a new geometry is defined and the whole process is repeated from step 2.

4.1.2. Sizing optimization

In sizing optimization problems the aim is usually to minimize the weight of the structure under certain behavioral constraints on stress and displacements. The design variables are most frequently chosen to be dimensions of the cross-sectional areas of the members of the structure. 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 fabrication limitations the design variables are not continuous but discrete since cross-sections belong to a certain set.

A discrete structural optimization problem can be formulated in the following form:

$$\begin{array}{lll} \text{min} & F(s), & \\ \text{subject to} & h_j(s) \le 0, & j = 1, \ldots, m \\ & s_i \in R^{\mathrm{d}}, & i = 1, \ldots, n \end{array} \tag{17}$$

where R^{d} is a given set of discrete values and design variables s_i $(i = 1, \ldots, n)$ can take values only from this set. In the present study the sizing optimization of multi-storey 3-D frames is investigated. Optimal designs of frames have been studied initially using conventional plastic design methods. Then more sophisticated optimization algorithms were introduced in order to solve this type of problems more efficiently [24,25]. Most frequently the objective function is the weight of the structure and the constraints are the member stresses and nodal displacements or inter-storey drifts. For rigid frames in rolled W-shapes, under allowable stress design requirements specified by Eurocode 3 [26], the stress constraints are the non-dimensional ratio q of the following formulas:

$$q = \frac{f_{\mathrm{a}}}{F_{\mathrm{a}}} + \frac{f_{\mathrm{b}}^{y}}{F_{\mathrm{b}}^{y}} + \frac{f_{\mathrm{b}}^{z}}{F_{\mathrm{b}}^{z}} \le 1.0, \quad \text{if} \quad \frac{f_{\mathrm{a}}}{F_{\mathrm{a}}} \le 0.15 \tag{18}$$

and

$$q = \frac{f_{\mathrm{a}}}{F_{\mathrm{a}}} + \frac{C_{\mathrm{m}} f_{\mathrm{b}}^{y}}{(1 - f_{\mathrm{a}}/F'_{\mathrm{e}})F_{\mathrm{b}}} + \frac{C_{\mathrm{m}} f_{\mathrm{b}}^{z}}{(1 - f_{\mathrm{a}}/F'_{\mathrm{e}})F_{\mathrm{b}}} \le 1.0, \quad \text{if} \quad \frac{f_{\mathrm{a}}}{F_{\mathrm{a}}} > 0.15 \tag{19}$$

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, $F_{\mathrm{a}} = 0.60 \times \sigma_y$ is the allowable axial stress,

$F_\mathrm{b} = 0.66 \times \sigma_y$ 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.

4.2. Reliability-based structural optimization

In reliability-based sizing optimization of large-scale multi-storey 3-D frames 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 variables are chosen to be the cross-sectional dimensions of structural members and the material properties, modulus 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

$$\begin{array}{lll} \min & F(s), & \\ \text{subject to} & h_j(s) \le 0, & j = 1, \ldots, m \\ & s_i \in R^\mathrm{d}, & i = 1, \ldots, n \\ & & p_\mathrm{f} \le p_\mathrm{a} \end{array} \tag{20}$$

where $F(s)$ is the objective function, s the vector of geometric design variables, $h_j(s)$ are the deterministic constraints, while p_f is the probability of failure of the structure required to remain below a threshold value (p_a) which comprise the probabilistic constraint. For rigid frames with W-shape cross-sections the stress constraints are also expressed by Eqs. (18) and (19). The design variables are selected from the W-shape database comprising of 311 elements.

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. (20).
3. The optimization phase is carried out with evolution strategies 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 [27,28]. 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 [12].

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 DBO and RBO problems. ES were introduced in the seventies by Rechenberg and Schwefel [29] and have three characteristics that make them differ from other conventional optimization algorithms: (i) in place of the usual deterministic operators, they use randomized 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.

5.1. ES in structural optimization problems

The ES optimization procedure starts with a set of parent vectors and if any of these parent vectors corresponds to 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 standard implementation of 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 reject $s_j$ 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.
```

The procedure is terminated as soon as the mean value of the objective values from all parent vectors in the last $2n\mu/\lambda$ generations has been improved by less than 0.01%.

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 FE analyses needed for the individuals of the 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 $\mu = \lambda$.

Although evolutionary type of algorithms were initially developed to solve unconstrained optimization problems, during the last decade several methods have been proposed for handling constrained optimization problems as well. The methods based on the use of penalty functions are employed in the majority of cases for treating constraint optimization problems. In a recent work by the authors [30] it has been observed that death penalty method employed by ES performs well for the problems considered. An extensive review of the constraint handling methods can be found in [31].

5.2. Deterministic-based structural optimization using ES and NN

After the selection of the suitable NN architecture and the training procedure is performed over a number (M) of data sets. In order to alleviate any inaccuracies entailed by the NN-based structural analysis a correction on the output values is proposed [32], especially when the constraint value is near the limit which divides the feasible with the infeasible region. Thus a relaxation of this limit is introduced before entering the optimization procedure during the NN testing phase. Therefore, a "correction" of the allowable constraint values is performed analogous to the maximum testing error of the NN configuration. The maximum testing error is the bigger average error of the output values among testing patterns. When the predicted values are smaller than the accurate ones derived from the conventional FE analysis then the allowable values of the constraints are decreased according to the maximum testing error of the NN configuration and vice versa. Li et al. [33] have implemented a hybrid methodology combining genetic algorithms and neural networks in predicting the dynamic response of plane frames, while El-Beltagy and Wright [34] have also proposed a combined GA and NN procedure, using Gaussian processes for improving generalization.

The combined ES-NN optimization procedure is performed in two phases. The first phase includes the training set selection, the FE analyses required to obtain the necessary I/O data for the NN training, and finally the selection, training and testing of a suitable NN configuration. The second phase is the ES optimization stage where instead of the conventional FE analyses the trained NN is used to predict the response of the structure in terms of objective and constraint functions' values due to different sets of design variables.

The proposed methodology ES-NN can be described with the following algorithm:

- NN training phase:
  1. *Training set selection step*: select M input patterns.
  2. *Constraints check*: perform the check for each input pattern vector.
  3. *Training step*: training of the NN.
  4. *Testing step*: test the trained NN.
- ES-NN optimization phase:
  1. *Parents initialization*.
  2. *NN constraints check*: all parent vectors become feasible.
  3. *Offspring generation*.
  4. *NN Constraints check*: if satisfied continue, else and go to step 3.
  5. *Parents' selection step*.
  6. *Convergence check*.

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 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. Monte Carlo simulation 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 behavior 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. Monte Carlo simulation 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. Monte Carlo simulation 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

The performance of the optimization methodologies discussed in previous sections is investigated in a number of characteristic test examples in sizing as well as in shape structural optimization, in deterministic as well as in reliability based optimization problems. One shape optimization benchmark test example and one characteristic 3-D building frame have been considered in order to illustrate the efficiency of the proposed methodologies for deterministic-based optimization problems. In the tables containing the results for the deterministic-based optimization problems the following abbreviations are used: *ES* refers to the standard implementation of the evolution strategies optimization procedure, in which structural analyses are performed using the conventional FE analysis. *ES-NN* refers to the combined ES and NN methodology, where the FE analysis response is predicted by a trained NN. For the two different types of training set selection that have been compared in this study the following abbreviations are used: (i) *UT* stands for the automatic selection of training set based on a uniform distribution of the design variables in the design space, while (ii) *GT* stands for the automatic selection of training set based on a Gaussian distribution of the design variables in the design space. The symbol "(c)" denotes that the allowable limits of the constraints have been adjusted, as discussed previously, in order to "correct" the NN predictions near the feasible region limits, while symbol "(v)" indicates that the final design is violating the constraints and thus it is infeasible. The symbols sd and ml stand for the minimization algorithms used during the training process, corresponding to steepest descent and the Levenberg–Marquard, respectively, while "(g)" means generalization. Only in the first test example we have examined the influence of the Levenberg–Marquard and the generalization.

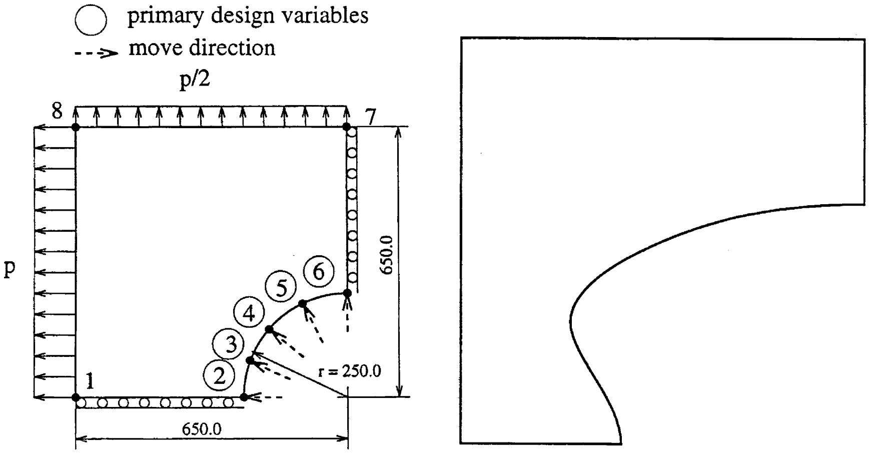
One characteristic 3-D building frame has been tested in order to illustrate the efficiency of the proposed methodologies for reliability-based sizing optimization problems. The probabilistic constraint is imposed on the probability of structural collapse due to successive formation of plastic nodes and is set to $p_{\mathrm{a}} = 0.001$. The probability of failure caused by uncertainties related to material properties, geometry and loads of the structures is estimated using MCS with the Importance Sampling 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 importance sampling function $g_x(x)$ for the loads is assumed to follow a normal distribution. In the tables containing the results for the reliability based optimization test example the following abbreviations are used: *DBO* stands for the standard deterministic-based optimization approach, *RBO* stands for the standard reliability-based optimization approach, while *RBO–NNi* corresponds to the proposed reliability-based optimization with NN incorporating algorithm i $(i = 1, 2)$.

6.1. Deterministic-based structural optimization test examples

6.1.1. Square plate with central cut-out problem

In this example plane stress conditions and isotropic material properties are assumed (elastic modulus $E = 210{,}000\,\mathrm{N/mm^2}$ and Poisson's ratio $\nu = 0.3$). The problem definition of this example is given in Fig. 1a, where due to symmetry only a quarter of the plate is modeled. The optimized shape is depicted in Fig. 1b. The two exterior sides of the plate are loaded with a distributed loading $p = 0.65\,\mathrm{N/mm^2}$, as shown in Fig. 1a. The objective is to minimize the volume of the structure subject to a limit to the equivalent stress $\sigma_{\max} = 7.0\,\mathrm{N/mm^2}$. The model consists of *eight* key points and *five* primary design variables (2, 3, 4, 5, 6) which can move along radial lines. The movement directions of the design variables are indicated by the dashed arrows. The stress constraints are imposed as a global constraint for all the Gauss points and as key point constraints for key points 2, 3, 4, 5, 6 and 8. For this test example the $(\mu + \lambda)$-ES approach is used with $\lambda = \mu = 5$.

The number of NN input units is taken equal to the number of design variables, the seven output units correspond to the seven constraint function values, whereas the NN architecture used has one hidden layer. The 5-11-7 NN architecture is therefore chosen and used for the optimization runs. Table 1 depicts the performance of the proposed ES-NN methodology, for all types of training set selection schemes, compared to the conventional ES optimization procedure for various numbers of NN training patterns. It is verified that the "correction" scheme of the allowable constraint values is more robust than the standard ES-NN version in producing feasible optimum designs. In addition it can be observed that a reduction in the number of training sets deteriorates the efficiency of the uniform type of training set selection, whereas for the Gaussian type it improves the efficiency with minor violations of the constraints in the final design. For this test case a significant improvement in total computing time required by ES-NN over ES is observed.



(a)

(b)

Fig. 1. Square plate: (a) initial shape; (b) final shape.

6.1.2. *Six storey space frame*

A six-storey space frame has been considered to illustrate the efficiency of the proposed methodology in deterministic-based sizing optimization problems with discrete design variables. In this example, the modulus of elasticity is 200 GPa (29,000 ksi) and the yield stress is $\sigma_y = 250$ MPa (36 ksi). The cross section of each member is assumed to be a W-shape and for each member two design variables are considered as shown in Fig. 2. The objective function of the problems is the weight of the structure. The constraints are imposed on the inter-storey drifts and the maximum non-dimensional ratio q of Eqs. (18) and (19) in each element group which combines axial force and bending moments. The values of allowable axial and bending stresses are $F_a = 150$ MPa and $F_b =$

Table 1
Square plate—performance of the optimization methods

Optimizer type	Optimization steps/ training patterns	ES-NN "steps"	Computing time (s)				Optimum volume (mm^3)
			Analysis	Training	ES-NN	Total	
ES	127/–	–	–	–	–	2141	279(v)
ES-NN-UT-sd	–/50	112	946.3	245.1	1.8	1193	272(v)
ES-NN-UT-sd(c)	–/50	132	946.3	245.1	2.0	1193	281(v)
ES-NN-UT-ml	–/50	112	946.3	203.2	1.8	1151	272(v)
ES-NN-UT-ml(g)	–/50	120	946.3	207.6	2.0	1156	280(v)
ES-NN-UT-sd	–/40	138	674.4	238.8	1.9	915	259(v)
ES-NN-UT-sd(c)	–/40	151	674.4	238.8	2.1	915	271(v)
ES-NN-UT-ml	–/40	138	674.4	191.5	1.9	868	259(v)
ES-NN-UT-ml(g)	–/40	123	674.4	194.9	1.8	871	280(v)
ES-NN-GT-sd	–/50	116	946.3	254.4	1.9	1203	270(v)
ES-NN-GT-sd(c)	–/50	125	946.3	254.4	2.1	1203	283(v)
ES-NN-GT-ml	–/50	116	946.3	212.9	1.9	1161	270(v)
ES-NN-GT-ml(g)	–/50	125	946.3	217.1	2.1	1166	279
ES-NN-GT-sd	–/40	112	674.4	238.8	1.8	915	263(v)
ES-NN-GT-sd(c)	–/40	132	674.4	238.8	2.0	915	275(v)
ES-NN-GT-ml	–/40	112	674.4	192.6	1.8	869	263(v)
ES-NN-GT-ml(g)	–/40	130	674.4	197.4	2.0	874	274(v)

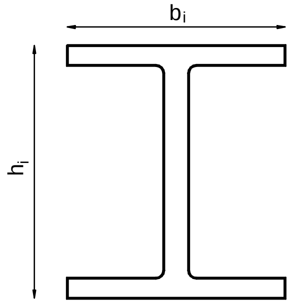


Fig. 2. W-shape cross-section design variables.

165 MPa, respectively, whereas the maximum allowable inter-storey drift is limited to 5.6 cm which corresponds to 1.5% of the height of each storey.

This example consists of 63 elements with 180 degrees of freedom as shown in Fig. 3a. The beams have length $L_1 = 7.32$ m and the columns $L_2 = 3.66$ m. The structure is loaded with a 19.16 kPa gravity load on all floor levels and a lateral load of 109 kN applied at each node in the front elevation along the *z*-direction. The element members are divided into five groups, as shown in Fig. 3b, each one having two design variables resulting in total to 10 design variables. The constraints are imposed on the maximum allowable inter-storey drift and the non-dimensional ratio *q* in each element group. For this test case the $(\mu+\lambda)$-ES approach is used with $\mu = \lambda = 5$.

For this example the number of NN input units is equal to the number of design variables, whereas the eleven output units correspond to the two values of axial force and bending moment for the five element groups, plus one for the maximum value of the inter-storey drift. The NN architecture used has one hidden layer, the 10-18-11 NN architecture is chosen and used for the optimization runs, while the minimization algorithm used is the steepest descent with no

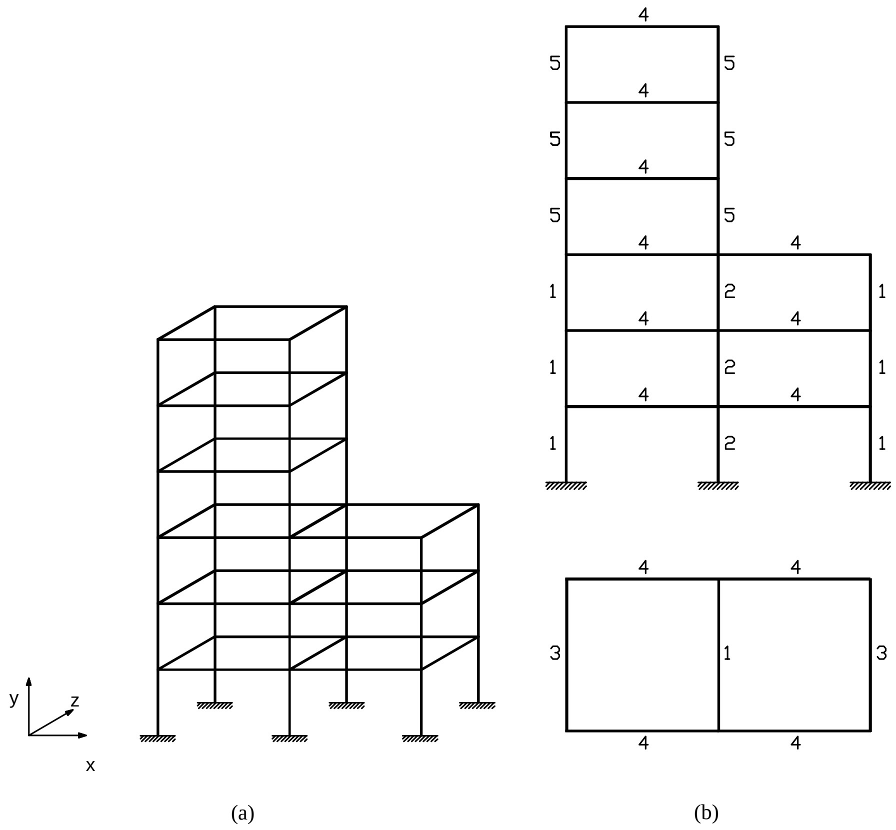


Fig. 3. (a) Six storey space frame; (b) element groups.

Table 2
Six storey space frame—performance of the optimization methods

Analysis type	Number of FE analysis/ training patterns	Number of NN analyses	Computing time (s)				Optimum weight (kN)
			Analysis	Training	ES-NN	Total	
ES	281/–	–	–	–	–	116.3	867
ES-NN-UT-ml	–/100	263	39.3	513.1	2.7	555.1	786(v)
ES-NN-UT-ml(c)	–/100	255	39.3	513.1	2.6	555	918(v)
ES-NN-UT-ml	–/80	187	33.1	342.6	1.8	377.5	685(v)
ES-NN-UT-ml(c)	–/80	232	33.1	342.6	2.4	378.1	758(v)
ES-NN-GT-ml	–/100	274	39.3	478.0	3.0	520.3	927(v)
ES-NN-GT-ml(c)	–/100	269	39.3	478.0	2.9	520.2	898(v)
ES-NN-GT-ml	–/80	274	33.1	321.7	3.0	357.8	940(v)
ES-NN-GT-ml(c)	–/80	266	33.1	321.7	2.9	357.7	884(v)
ES-NN-GT-ml	–/60	169	24.8	295.3	1.7	321.8	996(v)
ES-NN-GT-ml(c)	–/60	211	24.8	295.3	2.1	322.2	961(v)

generalization. Table 2 depicts, the performance of the proposed ES-NN methodology compared to the conventional ES optimization procedure for various numbers of NN training patterns. It can be observed that a marginal reduction in the number of training sets deteriorates the efficiency of the uniform type of training set selection, whereas it improves the efficiency of the Gaussian type with a minor increase of the objective function value in the final design. For this test case a slight increase in total computing time required by ES-NN over ES is observed due to the small size of the FE model of the structure.

6.2. *Reliability-based structural optimization test example*

6.2.1. *Twenty-storey space frame*

A 20-storey space frame has been considered to illustrate the efficiency of the proposed methodology in reliability based sizing optimization problems with discrete design variables. In this example the modulus of elasticity is 200 GPa (29,000 ksi) and the yield stress is $\sigma_y = 250$ MPa (36 ksi). The 20-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 front elevation of the frame. 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. (18) and (19) 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 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, 2 for the stresses of each element group and 1 for the inter-storey drift. The type of probability density functions, mean values, and variances of the random parameters are shown in Table 3. 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.

Table 3
Twenty storey space frame—characteristics of the random variables

Random variable	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

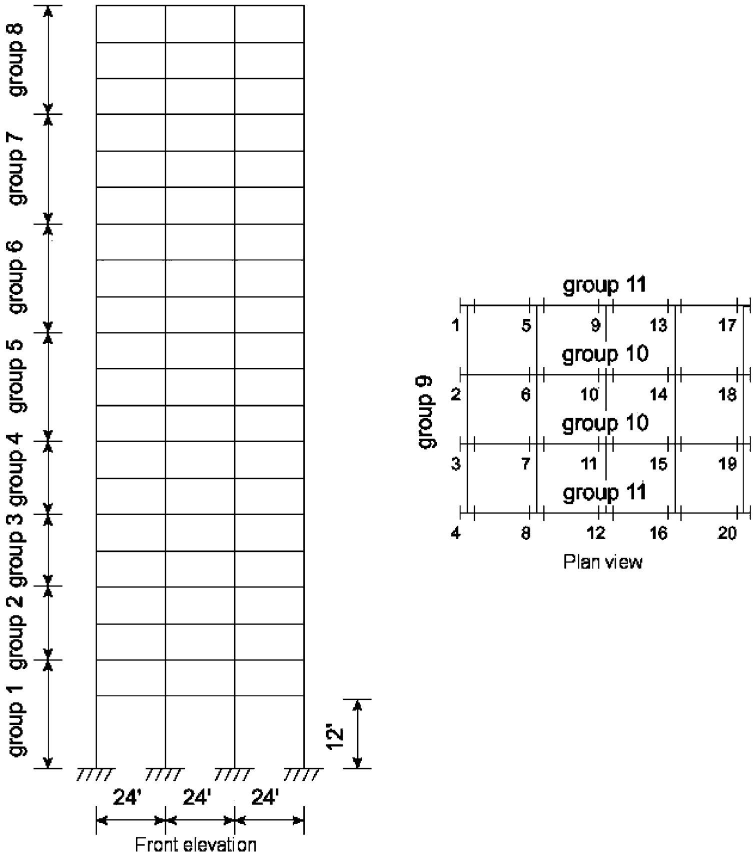


Fig. 4. Description of the 20-storey frame.

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 4 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 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 4. For both versions of the proposed methodology the minimization algorithm used is the steepest descent with no generalization. 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, using the uniform distribution selection of the training set.

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 4 the proposed RBO-NN2 optimization scheme manages to achieve the optimum weight in one tenth of the CPU time required by the

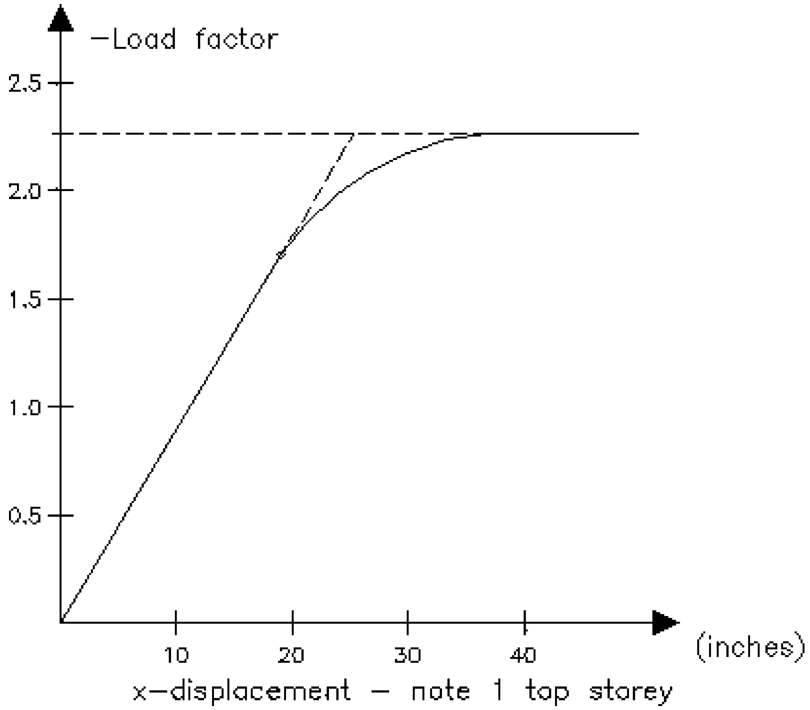


Fig. 5. Load–displacement curve.

Table 4
Twenty storey space frame—performance of the methods

Optimization procedure	ES generations	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 simulations)	126	0.103×10^{-2}	9114	141.0	28.4	14.1	7.1
RBO-NN1 (500 simulations)	129	0.102×10^{-2}	9121	34.5	7.2	3.5	1.8
RBO-NN2 (500 simulations)	126	0.103×10^{-2}	9114	15.8	3.3	1.7	0.9
RBO (1000 simulations)	120	0.103×10^{-2}	9156	250.3	50.1	25.1	12.6
RBO-NN1 (1000 simulations)	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.

conventional RBO procedure in sequential computing implementation.

Table 4 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 Monte Carlo simulations 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

The implementation of a hybrid optimization procedure, based on the combination of evolution strategies and neural networks, in deterministic and reliability based structural optimization problems was found to be very effective. The time-consuming requirements of repeated finite element analyses associated with the deterministic based optimization procedure using evolution strategies motivated the use of neural networks. The computational effort involved in the optimization procedure becomes excessive in large-scale problems and the use of neural networks to predict the necessary optimization data for evolution strategies can practically eliminate any limitation on the size of the problem, while the predicted structural response corresponding to different optimization simulations falls within acceptable tolerances.

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 70 times more expensive than the corresponding deterministic optimization procedure. The goal of decreasing the computational cost by one order of magnitude in sequential computing 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 computing mode with the proposed NN methodologies.

The methodology presented is an efficient, robust and generally applicable optimization procedure capable of finding the global optimum design of complicated structural optimization problems. Additionally, it was found that the proposed hybrid optimization methodology can reach the optimum for large and computationally intensive problems at a fraction of the computing time required by the standard implementation of evolution strategies optimization algorithm and the conventional method based on mathematical programming technique.

References

[1] S.D. Ragan, Sizing, shape, and topology design optimization of trusses using genetic algorithm, J. Struct. Eng. 121 (10) (1995) 1480–1487.

[2] M.C. Sharatchandra, M. Sen, M. Gad-el-Hak, New approach to constrained shape optimization using genetic algorithms, AIAA J. 36 (1) (1998) 51–61.

[3] M.B. Anderson, J.E. Burkhalter, R.M. Jenkins, Missile aerodynamic shape optimization using genetic algorithms, J. Spacecraft Rockets 37 (5) (2000) 663–669.

[4] S.Y. Woon, O.M. Querin, G.P. Steven, Structural application of a shape optimization method based on a genetic algorithm, J. Struct. Multidisc. Optim. 22 (1) (2001) 57–64.

[5] M. Papadrakakis, Y. Tsompanakis, N.D. Lagaros, Structural shape optimization using evolution strategies, Eng. Optim. J. 31 (1999) 515–540.

[6] J. Keane, Experiences with optimizers in structural design, in: I.C. Parmee (Ed.), Proceedings of the Conference on Adaptive Computing in Engineering Design and Control 94, Plymouth, 1994, pp. 14–27.

[7] H. Adeli, N.T. Cheng, Concurrent genetic algorithms for optimization of large structures, J. Aerospace Eng., ASCE 7 (3) (1994) 276–296.

[8] M. Papadrakakis, N.D. Lagaros, G. Thierauf, J. Cai, Advanced solution methods in structural optimization based on evolution strategies, J. Eng. Comput. 15 (1) (1998) 12–34.

[9] D.W. Coit, A.E. Smith, Penalty guided genetic search for reliability design optimization, Comput. Ind. Eng. 30 (4) (1996) 895–904.

[10] Y. Tsompanakis, M. Papadrakakis, Large-scale reliability-based structural optimization, J. Struct. Multidisc. Optim. (2003), in press.

[11] D.M. Frangopol, F. Moses, Reliability-based structural optimization, in: H. Adeli (Ed.), Advances in Design Optimization, Chapman & Hall, London, 1994, pp. 492–570.

[12] M. Papadrakakis, V. Papadopoulos, A computationally efficient method for the limit elasto plastic analysis of space frames, Comput. Mech. 16 (2) (1995) 132–141.

[13] 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.

[14] D. Frangopol, Interactive reliability based structural optimization, Comp. Struct. 19 (4) (1984) 559–563.

[15] C.G. Bucher, Adaptive sampling—an iterative fast Monte Carlo procedure, Struct. Safety 5 (1988) 119–126.

[16] M.I. Jordan, C.M. Bishop, Neural networks, MIT AI Lab, A.I. Memo No. 1562, C.B.C.L. Memo No. 131, 1996.

[17] D.J.C. MacKay, A practical Bayesian framework for back-propagation networks, Neur. Comput. 4 (2) (1992) 448–472.

[18] F.D. Foresee, M.T. Hagan, Gauss-Newton approximation to Bayesian regularization, in: Proceedings of the 1997 International Joint Conference on Neural Networks, 1997, pp. 1930–1935.

[19] O.C. Zienkiewicz, J.S. Cambell, Shape optimization and sequential linear programming, in: R.H. Galagher, O.C. Zienkiewicz (Eds.), Optimum Structural Design—Theory and Applications, Wiley, Chichester, 1973, pp. 109–125.

[20] A. Francavilla, C.V. Ramakrishnan, O.C. Zienkiewicz, Optimization of shape to minimize stress concentration, Int. J. Num. Meth. Eng. 10 (1975) 63–70.

[21] J. Haslinger, P. Neittaanmäki, Finite Element Approximation for Optimal Shape, Material and Topology Design, second ed., Wiley, Chichester, 1996.

[22] R.A.E. Mäkinen, J. Toivanen, J. Périaux, Multidisciplinary shape optimization in aerodynamics and electromagnetics using genetic algorithms, Int. J. Num. Meth. Fluids 30 (1999) 149–159.

[23] R.A.E. Mäkinen, T. Rossi, J. Toivanen, A moving mesh fictitious domain approach for shape optimization problems, M2AN Math. Model. Num. Anal. 34 (2000) 31–45.

[24] V. Thevendran, N.C. Das Gupta, G.H. Tan, Minimum weight design of multi-bay multi-storey steel frames, Comp. Struct. 43 (3) (1992) 495–503.

[25] G.M. Barsan, Optimal design of planar frames based on structural performance criteria, Comp. Struct. 3 (2) (1994) 1395–1400.

[26] Eurocode 3, Design of Steel Structures, Part 1.1: General Rules for Buildings, CEN, ENV 1993-1-1/1992.

[27] Y. Ueda, T. Yao, The plastic node method: a new method of plastic analysis, Comp. Meth. Appl. Mech. Eng. 34 (1982) 1089–1104.

[28] J.G. Orbinson, W. McGuire, J.F. Abel, Yield surface applications in non-linear steel frames analysis, Comp. Meth, Appl. Mech. Eng. 33 (1982) 557–573.

[29] H.P. Schwefel, Numerical Optimization for Computer Models, Wiley, Chichester, UK, 1981.

[30] N.D. Lagaros, M. Papadrakakis, G. Kokossalakis, Structural optimization using evolutionary algorithms, Comput. Struct. 80 (7–8) (2002) 571–587.

[31] C.A.C. Coello, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state-of-the-art, Comp. Meth., Appl. Mech. Eng. 191 (11) (2002) 1245–1287.

[32] M. Papadrakakis, N.D. Lagaros, Y. Tsompanakis, Structural optimization using evolution strategies and neural networks, Comput. Meth. Appl. Mech. Eng. 156 (1998) 309–333.

[33] Q.S. Li, D.K. Liu, A.Y.T. Leung, N. Zhang, C.M. Tam, L.F. Yang, Modeling of structural response and optimization of structural control system using neural network and genetic algorithm, The Struct. Design of Tall Build. 9 (2000) 279–293.

[34] M. El-Beltagy, W. Wright, Gaussian processes for model fusion, in: G. Dorffner, H. Bischof, K. Hornik (Eds.), ICANN 2001, LNCS 2130, 2001, pp. 376–383.