

Design optimization of steel structures considering uncertainties

M. Papadrakakis*, N.D. Lagaros, V. Plevris

Institute of Structural Analysis & Seismic Research, National Technical University of Athens, Zografou Campus, Athens 157 80, Greece

Received 10 January 2005; received in revised form 7 April 2005; accepted 7 April 2005

Available online 17 May 2005

Abstract

In real world engineering applications the uncertainties of the structural parameters are inherent and the scatter from their nominal ideal values is in most cases unavoidable. These uncertainties play a dominant role in structural performance and the only way to assess this influence is to perform Reliability-Based Design Optimization (RBDO) and Robust Design Optimization (RDO). Compared to the basic deterministic-based optimization problem, a RBDO problem considers additional non-deterministic constraint functions, while the RDO yields a design with a state of robustness, so that its performance is the least sensitive to the variability of the uncertain parameters. The first part of this study examines the application of Neural Networks (NN) to the RBDO of large-scale structural systems, while the second part investigates the structural RDO problem. The use of NN in the framework of the RBDO problem is motivated by the approximate concepts inherent in reliability analysis and the time-consuming repeated analyses required by Monte Carlo Simulation. On the other hand the RDO is a multi-criteria optimization problem where the aim is to minimize both the weight of the structure and the variance of the structural response.

© 2005 Elsevier Ltd. All rights reserved.

Keywords: Structural optimization; Reliability analysis; Robust design; Evolution Strategies; Monte Carlo simulation; Neural Networks

1. Introduction

The development of stochastic analysis methods that has taken place during the last two decades [1] has stimulated interest in the probabilistic optimum design of structures. Reliability-Based Design Optimization (RBDO) [2–4] and Robust Design Optimization (RDO) [5–7] are two distinct design formulations that account for probabilistic system response. Despite the theoretical advances in the field of reliability analysis, serious computational obstacles arise when treating realistic problems. In particular, the reliability-based design optimization of large-scale structural systems is an extremely computationally intensive task, as shown by Tsompanakis and Papadrakakis [8]. Despite the improvements achieved in the efficiency of the computational methods for treating reliability analysis

problems, they still require disproportionate computational effort for practical applications. This is why very few successful numerical investigations are known in this field.

In the first part of the present study the reliability-based sizing optimization of multi-storey space frames is investigated. The main goal of RBDO methods is to design for safety with respect to extreme events. The objective function is the weight of the structure while the constraints are both deterministic (as regards stresses and displacements) and probabilistic (the overall probability of failure of the structure). Uncertainty of loads, material properties and member geometry are taken into account in the reliability analysis using the Monte Carlo simulation (MCS) method. The probability of failure of the structure is determined via a limit state elasto-plastic analysis. Two methodologies that combine Evolution Strategies and Neural Networks (ES–NN) are examined. In the first, a trained NN is applied to predict the response of the structure in terms of deterministic and probabilistic constraint checks due to different sets of design variables. In the second methodology, the limit state elasto-plastic analyses required

* Corresponding author. Tel.: +30 210 772 1694; fax: +30 210 772 1693.

E-mail addresses: mpapadra@central.ntua.gr (M. Papadrakakis), nlagaros@central.ntua.gr (N.D. Lagaros), vplevris@central.ntua.gr (V. Plevris).

for the MCS method are replaced by the NN predictions 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 that utilizes available information generated from selected conventional elasto-plastic analyses. The trained NN is used to predict the critical load factor due to different sets of basic random variables.

In the second part of the study the robust design sizing optimization of steel space trusses is examined. RDO methods primarily seek to minimize the influence of stochastic variations on the mean design. The objective functions considered are the weight and the standard deviation of the response of the structure, subject to stress and displacement constraints imposed by the design codes. Uncertainties of loads, material properties and member geometry are taken into account in the stochastic analysis using the MCS method. Each design is checked for whether it satisfies the constraints with a prescribed probability of violation. The optimization problem at hand is a multi-criterion optimization problem. Evolutionary algorithms, and in particular the Evolution Strategies method, are implemented to perform the optimization phase.

The paper is organized as follows. A presentation of the deterministic-based design optimization problem with one or multiple objective functions is given in Section 2. The reliability-based design optimization problem is described in Section 3, while the next section deals with the robust design optimization problem. Subsequently, Section 5 presents the Monte Carlo simulation method applied for stochastic analysis. The Neural Networks and the Evolution Strategies method are described in Sections 6 and 7, respectively. The test examples that demonstrate the potential of the proposed approaches in solving realistic problems are presented in Section 8, followed by concluding remarks in Section 9.

2. Deterministic-based design optimization

2.1. Single-objective sizing optimization problem

In single-objective deterministic-based sizing optimization problems the aim is to minimize the weight of the structure under certain deterministic behavioural constraints usually on stresses and displacements. Due to engineering practice demands, the members of a frame or a truss structure are divided into groups, with the members of each group sharing the same design variables. This linking of elements results in a trade-off between the use of more material and the need for symmetry and uniformity of structures due to practical considerations. Furthermore, it has to be taken into account that in most practical sizing optimization cases, due to manufacturing limitations the design variables cannot be considered as continuous but should be treated as discrete since cross sections belong to a certain set provided by the manufacturers.

A discrete Deterministic-Based Optimization (DBO) problem can be formulated in the following form:

$$\begin{aligned} \min \quad & F(\mathbf{s}) \\ \text{subject to} \quad & g_j(\mathbf{s}) \leq 0 \quad j = 1, \dots, k \\ & s_i \in R^d, \quad i = 1, \dots, n \end{aligned} \quad (1)$$

where $F(\mathbf{s})$ is the objective function, $\mathbf{s}$ is the vector of design variables, which can take values only from a given discrete set R^d , and $g_j(\mathbf{s})$ are the deterministic constraints. Most frequently the deterministic constraints refer to member stresses and nodal displacements or inter-storey drifts.

2.2. Multi-objective sizing optimization problem

In practical applications of sizing optimization problems, the material weight itself does not always correspond to a representative measure of the structural performance. In fact, several conflicting and usually incommensurable criteria usually exist in real-life design problems that have to be dealt with simultaneously. Thus the designer is forced to look for a good compromise among conflicting requirements. Problems of this kind are called multi-objective optimization problems. The consideration of such problems in their present form originated towards the end of the 19th century when Pareto [9] presented the optimality concept in economic problems with several conflicting criteria.

2.2.1. Criteria and conflict in multi-objective structural optimization

In structural optimization, the designer is faced with the question of choosing the most suitable criteria for measuring the economy, strength, serviceability or any other factor that may affect the performance of the structure. Any quantity that has an influence on the structural performance can be considered as a criterion. One important basic property in the multi-criterion formulation is the conflict that may exist among the various criteria. Only quantities that are competing with each other should be treated as independent criteria whereas the others can be combined into a single criterion that can represent the whole group. Two functions f_i and f_j are called locally collinear with no conflict at point $\mathbf{s}$ if there exists $c > 0$ such that $\nabla f_i(\mathbf{s}) = c \nabla f_j(\mathbf{s})$. Otherwise, the functions are called locally conflicting at point $\mathbf{s}$. According to this definition any two criteria are locally conflicting at a point of the design space if their maximum improvement is achieved in different directions. Furthermore, two functions f_i and f_j are called globally conflicting in the feasible region $\mathbf{F}$ of the design space when the two optimization problems $\min\{\mathbf{s} \in \mathbf{F} f_i(\mathbf{s})\}$ and $\min\{\mathbf{s} \in \mathbf{F} f_j(\mathbf{s})\}$ have different optimal solutions.

2.2.2. Formulation of a multi-objective optimization problem

In formulating an optimization problem, the choice of design variables, criteria and constraints represents undoubtedly the most important decision to be made by

the engineer. In general, the mathematical formulation of a multi-objective problem that includes a set of n design variables, a set of m objective functions and a set of k constraint functions can be defined as follows:

$$\begin{aligned} \min_{s \in \mathbf{F}} \quad & [f_1(s), f_2(s), \dots, f_m(s)]^T \\ \text{subject to} \quad & g_j(s) \leq 0 \quad j = 1, \dots, k \\ & s_i \in R^d, \quad i = 1, \dots, n \end{aligned} \quad (2)$$

where $\mathbf{F}$ is the feasible region, a subspace of the design space R^n for which the constraint functions $g(s)$ are satisfied:

$$\mathbf{F} = \{s \in R^n \mid g_j(s) \leq 0 \quad j = 1, \dots, k\}. \quad (3)$$

If the m objective functions are globally conflicting, there exists no unique point which would represent the optimum for all of them. Thus the common optimality condition used in single-objective optimization must be replaced by a new concept, the so-called *Pareto* optimum. A design vector $s^* \in \mathbf{F}$ is a Pareto optimum for the problem of Eq. (2) if and only if there is no other design vector $s \in \mathbf{F}$ such that

$$\begin{aligned} f_i(s) &\leq f_i(s^*) \text{ for } i = 1, \dots, m \\ \text{with } f_j(s) &< f_j(s^*) \text{ for at least one objective } j. \end{aligned} \quad (4)$$

These conditions mean that a point in the feasible design space is considered as Pareto optimal if there is no other point that would improve the value of even one objective function without degrading the value of at least one other. The set of Pareto optimum solutions represent the solution of the optimization problem with multiple objectives. In practical applications however, the designer seeks for a unique optimum solution to be implemented in practice; thus a compromise should be made among the available Pareto optimal solutions.

2.2.3. Solving the multi-objective optimization problem

Classical methods for generating the Pareto optimal set combine the objectives into a single, parameterized objective function. Several optimization runs with different parameter settings are performed in order to generate the Pareto optimal set. Representatives of this class of techniques are the linear weighting method, the constraint method, goal programming and the minmax approach. In this study the first method is adopted since this procedure is independent of the underlying optimization algorithm and was proved to be efficient [10].

In real world problems different measuring units correspond to different objective functions resulting in variations of some orders of magnitude among the values of the objectives. It is therefore advisable that the objectives be normalized according to the following expression:

$$\tilde{f}_i(s) = \frac{f_i(s) - f_{i \min}}{f_{i \max} - f_{i \min}} \quad (5)$$

where the normalized objectives $\tilde{f}_i(s) \in [0, 1]$, $i = 1, 2, \dots, m$, use the same design space as the non-normalized ones, while $f_{i \min}$ and $f_{i \max}$ are the minimum and maximum expected values of the i -th objective function.

The linear weighting method combines all the objectives into a single-scalar parameterized objective function by using weighting coefficients. If w_i , $i = 1, 2, \dots, m$ are the weighting coefficients, the problem of Eq. (2) can be written as follows:

$$\min_{s \in \mathbf{F}} \sum_{i=1}^m w_i \tilde{f}_i(s), \quad (6)$$

while with no loss of generality the following normalization of the weighting coefficients is employed:

$$\sum_{i=1}^m w_i = 1. \quad (7)$$

By varying the weights it is now possible to generate the set of Pareto optimum solutions for the problem of Eq. (2). The values of the weighting coefficients can be adjusted according to the importance of each criterion. Every combination of the weighting coefficients corresponds to a single Pareto optimal solution; thus by performing a set of optimization processes using different weighting coefficients it is possible to generate the full set of the Pareto optimal solutions.

3. Reliability-based design optimization

In sizing DBO problems the aim is to minimize the weight of the structure under certain deterministic behavioural constraints usually on stresses and displacements. In RBDO problems additional probabilistic constraints are imposed in order to take into account the influence of various random parameters. Probabilistic constraints define the feasible region of the design space by restricting the probability that a constraint is violated within an allowable probability of violation.

In the present study the reliability-based sizing optimization of real-scale multi-storey 3D frames is investigated. Thus the overall probability of failure of the structure, as a result of a limit state elasto-plastic analysis, is taken as the global reliability constraint. Such a constraint enforces the condition that the probability of failure of the structure is smaller than a certain specified value. The probabilistic design variables are chosen to be the cross sectional dimensions of the structural members and the material properties, modulus of elasticity E and yield stress σ_y .

A discrete RBDO problem can be formulated in the following form:

$$\begin{aligned} \min \quad & F(s) \\ \text{subject to} \quad & g_j(s) \leq 0 \quad j = 1, \dots, m \\ & s_i \in R^d, \quad i = 1, \dots, n \\ & p_f \leq p_a \end{aligned} \quad (8)$$

where $F(s)$ is the objective function, s is the vector of geometric design variables, which can take values only from a given discrete set R^d , $g_j(s)$ are the deterministic constraints and p_f is the probability of failure of the

structure, bounded to remain below a threshold value (p_a) that comprises the probabilistic constraint. The applied loading for examining the deterministic constraints is deterministically defined by the design codes, while in the case of the probabilistic constraint the applied loading is a random variable following a specific distribution. Most frequently, the deterministic constraints refer to member stresses and nodal displacements or inter-storey drifts.

The proposed RBDO sizing optimization methodology proceeds with the following steps:

1. At the outset of the optimization procedure the geometry, boundaries and reference loads of the structure under investigation are defined.
2. The constraints are defined.
3. The optimization phase is carried out with Evolution Strategies where feasible designs are produced at each generation. The feasibility of each design vector is checked with respect to both the deterministic and the 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 determined with a reliability analysis of the structure using the MCS technique for the evaluation of the probability of failure.
6. If the convergence criteria for the optimization algorithm are satisfied, then the optimum solution has been achieved and the process is terminated; otherwise 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 the structure. This failure criterion is considered to be the formation of a mechanism as a result of a limit state elasto-plastic analysis of the space frame structure without considering member instability effects. The incremental non-holonomic first-order step-by-step limit state analysis adopted is based on the generalized plastic node concept. 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 yield surface, internal and homothetic to the initial one, is implemented which forms a plastic zone for the activation of the plastic nodes.

4. Robust design optimization

In a robust design structural sizing optimization problem an additional objective function is considered which is related to the influence of the random nature of some structural parameters on the performance of the structure. In the present study the aim is to minimize both the weight and the standard deviation of the response of the structure. The constraint functions are also varied due to

variations of the uncertain structural parameters. A DBO optimum solution might violate some of the constraints for some values of the uncertain parameters. In the formulation of the RDO considered in this study the variance of the constraints has also been taken into account and additional constraint functions of stochastic nature are considered. The mathematical formulation of the RDO problem implemented in this study is as follows:

$$\begin{aligned} \min \quad & \Phi(\mathbf{s}) \\ \text{subject to} \quad & g_j(\mathbf{s}) \leq 0 \quad j = 1, \dots, k \\ & p_{v,j} \leq p_{\text{all}} \quad j = 1, \dots, k \\ & s_i \in R^d, \quad i = 1, \dots, n \end{aligned} \quad (9)$$

where $\Phi(\mathbf{s})$ is the multi-objective function, $\mathbf{s}$ is the vector of geometric design variables, which can take values only from a given discrete set R^d , $g_j(\mathbf{s})$ are the deterministic constraints while $p_{v,j}$ is the probability of violation of the j -th deterministic constraint bound by an upper allowable probability equal to p_{all} . Probabilistic constraints define the feasible region of the design space by restricting the probability that a deterministic constraint is violated within an allowable probability of violation. The multi-objective function is expressed as

$$\Phi(\mathbf{s}) = w\tilde{F}(\mathbf{s}) + (1 - w)\tilde{\sigma}_u \quad (10)$$

where $\tilde{F}(\mathbf{s})$ is the weight of the structure and $\tilde{\sigma}_u$ is the standard deviation of the response of the structure normalized according to Eq. (5). The proposed robust design sizing optimization methodology proceeds with the following steps:

1. At the outset of the optimization procedure the geometry, boundaries and reference loads of the structure under investigation are defined.
2. The constraints are defined.
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. Stochastic analysis of the structure using the MCS technique is carried out in order to evaluate the probability of violation of the constraints and to calculate the variance of the response of the structure.
6. If the convergence criteria for the optimization algorithm are satisfied, then the optimum solution has been achieved and the process is terminated; otherwise the whole process is repeated from step 3 with a new generation of design vectors.

5. Stochastic analysis

In stochastic analysis of structures the MCS method is particularly suitable when an analytical solution is not

attainable. This is mainly the case in problems of complex nature with a large number of random variables (random structural parameters), where all other stochastic analysis methods are not applicable. Despite the fact that the mathematical formulation of the MCS is simple and the method has the capability of handling practically every possible case regardless of its complexity, this approach has not met with overwhelming acceptance due to the excessive computational effort that it requires. However, soft computing methodologies and parallel processing have recently been implemented, having a beneficial effect on the efficiency of MCS [11,12].

In structural stochastic analysis problems where the probability of violation of some behavioural constraints, or any other statistical quantity, is to be calculated, the MCS approach can be stated as follows: Expressing the limit state function as $G(\mathbf{x}) < 0$, where $\mathbf{x} = [x_1, x_2, \dots, x_M]^T$ is a vector of the random structural parameters, the probability of violation of the behavioural constraints can be written as

$$p_{\text{viol}} = \int_{G(\mathbf{x}) \geq 0} f_{\mathbf{x}}(\mathbf{x}) d\mathbf{x} \quad (11)$$

where $f_{\mathbf{x}}(\mathbf{x})$ denotes the joint probability of violation for all random structural parameters. Since MCS is based on the theory of large numbers (N_{∞}) an unbiased estimator of the probability of violation is given by

$$p_{\text{viol}} = \frac{1}{N_{\infty}} \sum_{j=1}^{N_{\infty}} I(\mathbf{x}_j) \quad (12)$$

where $\mathbf{x}_j$ is the j -th vector of the random structural parameters, and $I(\mathbf{x}_j)$ is an indicator for successful and unsuccessful simulations defined as

$$I(\mathbf{x}_j) = \begin{cases} 1 & \text{if } G(\mathbf{x}_j) \geq 0 \\ 0 & \text{if } G(\mathbf{x}_j) < 0. \end{cases} \quad (13)$$

In order to estimate p_{viol} , an adequate number of N independent random samples is produced using a specific probability density function of the vector $\mathbf{x}_j$. The value of the violation function is computed for each random sample $\mathbf{x}_j$ and the Monte Carlo estimation of p_{viol} is given in terms of sample mean by

$$p_{\text{viol}} \cong \frac{N_H}{N} \quad (14)$$

where N_H is the number of successful simulations and N is the total number of simulations.

6. Neural Networks

Feed-forward Neural Networks consist of a number of units (neurons) linked together and attempt to create a desired relation in an input/output set of learning patterns. A learning algorithm tries to determine a set of weight parameters, in order to achieve the right response for each input vector applied to the network. The training procedure can be considered as an unconstrained optimization problem

where the goal is to find the vector/matrix $\mathbf{w}$ that minimizes the following real valued scalar function:

$$\mathbf{E} = \mathbf{E}(\mathbf{w}). \quad (15)$$

The numerical minimization algorithms used for the training generate a sequence of matrices of the weight parameters, through an iterative procedure [13]. To apply an algorithmic operator $\mathcal{H}$ we need a starting weight matrix $\mathbf{w}^{(0)}$, while the iteration formula can be written as follows:

$$\mathbf{w}^{(t+1)} = \mathcal{H}(\mathbf{w}^{(t)}) = \mathbf{w}^{(t)} + \Delta \mathbf{w}^{(t)}. \quad (16)$$

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

$$\Delta \mathbf{w}^{(t)} = a_t \mathbf{d}^{(t)} \quad (17)$$

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

The algorithm that was used for the NN training in this study is the Levenberg–Marquardt method with the following direction vector:

Levenberg–Marquardt method: $\mathbf{d}^{(t)} = -[\mathbf{H}(\mathbf{w}^{(t)}) + \lambda_t \mathbf{I}]^{-1} \nabla \mathbf{E}(\mathbf{w}^{(t)})$ where λ_t is a positive constant while $\mathbf{H}(\mathbf{w})$ and $\nabla \mathbf{E}(\mathbf{w}^{(t)})$ are the Hessian matrix and the gradient of the function $\mathbf{E}$, respectively.

$$\nabla \mathbf{E}(\mathbf{w}) = \mathbf{J}(\mathbf{w})^T \mathbf{E}(\mathbf{w}) \quad (18)$$

where $\mathbf{E}(\mathbf{w}) = [E_1(\mathbf{w}), \dots, E_m(\mathbf{w})]^T$ is the error vector.

$$\nabla^2 \mathbf{E}(\mathbf{w}) = \mathbf{H}(\mathbf{w}) = \mathbf{J}(\mathbf{w})^T \mathbf{J}(\mathbf{w}) + \sum_{i=1}^m E_i(\mathbf{w}) \mathbf{H}_i(\mathbf{w}) \quad (19)$$

where $\mathbf{J}(\mathbf{w})$ is the Jacobian matrix of the vector function $\mathbf{E}(\mathbf{w})$ and $\mathbf{H}_i(\mathbf{w})$ is the Hessian matrix of the component function $E_i(\mathbf{w})$.

7. Evolution Strategies

The initial versions of the Evolution Strategies (ES) method were based on a population consisting of one individual only. The two-membered ES scheme is the minimal concept for an imitation of the organic evolution. The two principles of mutation and selection, which Darwin in 1859 recognised as the most important, are taken as rules for variation of the design variables and for recursion of the iteration sequence, respectively. The multi-membered ES method employed in this study is based on a population consisting of more than one individual; this is the main difference from the previous two-membered methodology.

7.1. Evolution Strategies for structural multi-objective optimization problems

For the solution of multi-objective optimization problems a great number of methods have been proposed in the past [10,14,15]. In our implementation, the linear weighting

sum method is used in order to generate the set of Pareto optimal solutions; the optimization procedure initiates with a set of parent design vectors needed by the ES optimizer and a set of weighting coefficients for the combination of all objectives into a single scalar parameterized objective function. These weighting coefficients are not set by the designer but are varied by the optimizer after a Pareto optimal solution has been achieved. There is an outer loop, called the *decision making loop*, which systematically varies the weight coefficients of the objective function. The inner loop is the classical ES procedure, starting with an initial set of parent vectors. If any of the parent vectors gives an unfeasible design, then it is modified until it becomes feasible. Subsequently, the offspring are generated and checked for whether they are in the feasible region. According to the $(\mu + \lambda)$ selection scheme, in every generation the values of the objective function of the parent and the offspring vectors are compared and the worst vectors are rejected, while the remaining ones are taken as the parent vectors of the new generation. On the other hand, according to the (μ, λ) selection scheme only the offspring vectors of each generation are used to produce the new generation. This procedure is repeated until the termination criterion is satisfied. The multi-objective optimization ES algorithm combined with the standard methods can be stated as follows:

Outer loop—decision making loop

Set the coefficients w_i of the parameterized objective function

Inner loop—ES loop

1. *Selection step*: Selection of s_i ($i = 1, 2, \dots, \mu$) parent vectors of the design variables
2. *Analysis step*
3. *Evaluation of parameterized objective function*
4. *Constraints check*: All parent vectors become feasible
5. *Offspring generation*: Generate s_j ($j = 1, 2, \dots, \lambda$) offspring vectors of the design variables
6. *Analysis step*
7. *Evaluation of the parameterized objective function*
8. *Constraints check*: If satisfied continue; else go to step 5 and change s_j
9. *Selection step*: Selection of the next generation parents according to $(\mu + \lambda)$ or (μ, λ) selection schemes
10. *Convergence check*: If satisfied stop; else go to step 5

End of inner loop

End of outer loop

7.2. 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 constraint enforces 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 state 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). In this study two combined ES–NN methodologies are examined for the solution of the RBDO problem.

7.2.1. NN used for deterministic and probabilistic constraints check

In the first methodology, a trained NN that utilizes information generated from a number of properly selected design vectors is used to perform both the deterministic and probabilistic constraints checks that are needed 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 constraint 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. *Parent 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. *Parent selection step*
6. *Convergence check*

7.2.2. NN prediction of the critical load in structural failure

In the second methodology the limit state elasto-plastic analyses required during the MCS are replaced by the 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 are processed to obtain input and output pairs, which are used for the NN training. 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 state elasto-plastic analyses by predicted results obtained with a trained NN, a training procedure is performed based on data collected from a number of conventional limit state elasto-plastic analyses. After the completion of the training phase, the trained NN predictions replace the conventional limit state elasto-plastic analyses, for the current design. For the selection of suitable training pairs, the sample space for each random variable is divided into smaller intervals of equal size. 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. *Parent 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. *Parent selection step*
10. *Convergence check*

8. Test examples

8.1. Six-storey space frame—RBDO test example

A three-dimensional frame has been tested in order to illustrate the efficiency of the proposed methodologies for treating realistic large-scale reliability-based sizing optimization problems. The cross section of each member is

assumed to be a W-shape and one design variable is allocated for each member. The objective function is the weight of the structure to be minimized. The deterministic constraints are applied on the inter-storey drifts and the member stresses.

According to Eurocode 3 [16] for the design of steel structures, the interstorey drift constraint employed for a frame structure can be written as

$$\frac{d_r}{\nu} \leq 0.006 \times h \quad (20)$$

where ν is a reduction factor for the serviceability limit state (taken equal to 2.5 for this test example) and d_r is the relative drift of two consecutive stories.

The stress constraint function for beams subjected to biaxial bending under compression, for each group of structural members, is given by the formula

$$\frac{N_{sd}}{A f_y / \gamma_{M1}} + \frac{M_{sd,y}}{W_{pl,y} f_y / \gamma_{M1}} + \frac{M_{sd,z}}{W_{pl,z} f_y / \gamma_{M1}} \leq 1.0 \quad (21)$$

where N_{sd} , $M_{sd,y}$, $M_{sd,z}$ are the computed stress resultants, $W_{pl,y}$, $W_{pl,z}$ are the plastic first moments of inertia, f_y is the yield stress and γ_{M1} is a safety factor equal to 1.10 [16].

This example consists of 63 elements with 180 degrees of freedom as shown in Fig. 1. The structure is loaded with a gravity load of 19.16 kPa on all floor levels and a lateral load of 110 kN applied at each node in the front elevation along the x direction. The members of the structure are divided into five groups, as shown in Fig. 1, each one having one design variable. The deterministic constraints are eleven, two for the stresses of each element group and one for the inter-storey drift. The load–displacement curve of a node in the top floor of the frame is depicted in Fig. 2, corresponding to the design vector ($W_{12 \times 26}$, $W_{12 \times 33}$, $W_{12 \times 87}$, $W_{12 \times 87}$, $W_{10 \times 60}$) with a probability of failure equal to 8.68%. For this test example the $(\mu + \lambda)$ -ES approach is used with $\mu = \lambda = 5$ following the rule that μ , λ should be equal to the number of the design variables. Sample sizes of 500, 1000 and 5000 simulations have been examined for the MCS with the Importance Sampling (IS) technique, in order to study the influence of the number of simulations on the optimization process.

The probabilistic constraint is imposed on the probability of structural collapse due to successive formation of plastic nodes and is set to $p_a = 10^{-3}$. The probability of failure due to uncertainties related to material properties, geometry and loads of the structure is estimated using MCS with IS. External loads, yield stresses, elastic moduli and the dimensions of the cross sections of the structural members are considered as 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 also assumed to follow a normal distribution. The type of probability density function, mean value and variance of the random parameters are described in Table 1. Table 2 shows the performance of the methods for the

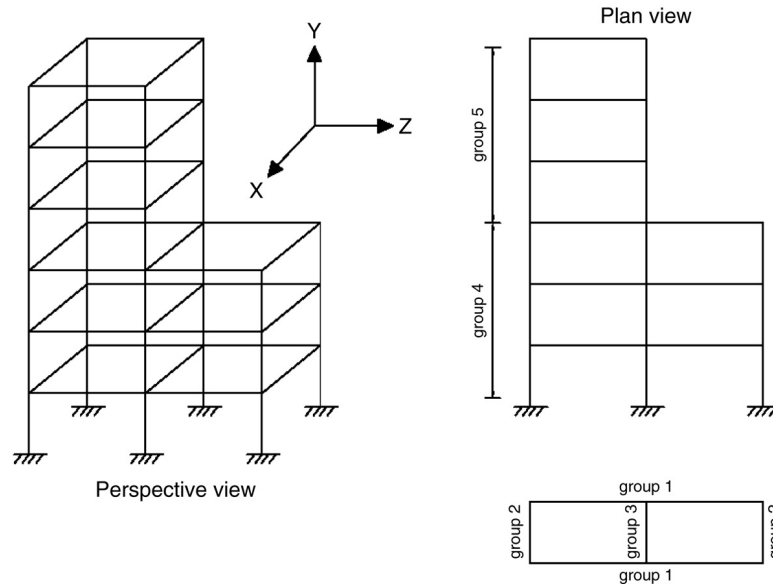


Fig. 1. 3D six-storey frame: description.

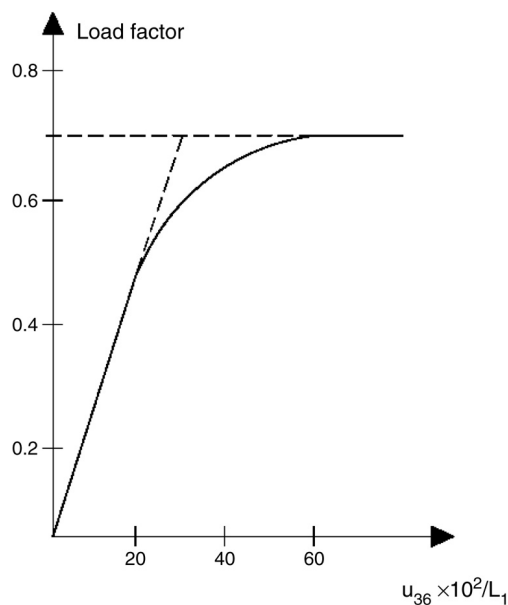


Fig. 2. 3D six-storey frame: load–displacement curve.

results of the test examples considered. *DBO* stands for the conventional Deterministic-Based Optimization approach, *RBDO* stands for the conventional Reliability-Based Design Optimization approach, while *RBDO-NNi* corresponds to the proposed Reliability-Based Optimization with the NN incorporating *algorithm i* ($i = 1, 2$).

As can be observed from Table 2 the probability of failure for the deterministic optimum is unacceptable since it substantially exceeds the accepted value of 10^{-3} . On the other hand, the optimum weight achieved by the RBDO exceeds the deterministic one by 16%. For the application of the RBDO-NN1 methodology the number of NN input

units is equal to the number of design variables, whereas one output unit is needed. The output unit takes the values of 1 or 0, corresponding to a feasible or unfeasible design vector. Consequently the NN configuration implemented in this case has one hidden layer with 10 nodes resulting in a 5–10–1 NN architecture used for all runs [12]. The training set consists of 100 training patterns chosen based on the requirement that the full range of the design space should be represented in the training procedure.

Table 1
3D six-storey frame: Characteristics of the random variables

Random variable	Probability density function	Mean value	Standard deviation
E	N	200	$0.10E$
σ_y	N	25.0	$0.10\sigma_y$
Design variables	N	s_i	$0.1s_i$
Loads ($\times 10^3$)	Log- N	6.4	0.20

For the application of the RBDO-NN2 methodology the number of NN input units is equal to the number of the random variables, whereas one output unit is needed corresponding to the critical load factor. Consequently the NN configuration results in a 3–7–1 NN architecture which is used for all runs [12]. The number of conventional step-by-step limit state analysis calculations performed for the training of the NN is taken as 60 corresponding to different groups of random variables properly selected from the random field. As can be seen in Table 2 the proposed RBDO-NN1 and RBDO-NN2 optimization schemes for 500 and 1000 simulations manage to achieve the optimum weight in one third of the CPU time required by the conventional

Table 2
3D six-storey frame: performance of the methods

Optimization procedure	ES generations	p_f^b	Optimum weight (kN)	Time (h)
DBO	43	0.171×10^{-0}	727	0.05
RBDO (500 siml.)	65	0.105×10^{-2}	869	7.6
RBDO-NN1 (500 siml.)	64	0.105×10^{-2}	873	2.7
RBDO-NN2 (500 siml.)	65	0.105×10^{-2}	869	3.6
RBDO (1000 siml.)	68	0.101×10^{-2}	875	16.3
RBDO-NN1 (1000 siml.)	69	0.101×10^{-2}	875	5.3
RBDO-NN2 ^a	66	0.97×10^{-3}	881	5.0
RBDO (5000 siml.)	68	0.101×10^{-2}	875	81.1
RBDO-NN1 (5000 siml.)	69	0.101×10^{-2}	875	26.5
RBDO-NN2 ^a	66	0.97×10^{-3}	881	5.0

^a For 100 000 simulations.

^b For 100 000 simulations using the NN2 scheme.

RBDO procedure. For 5000 simulations the time required for RBDO-NN1 remains one third of the conventional one, while for the RBDO-NN2 the CPU time required drops to one sixteenth. The number of MC simulations in the case of the NN2 scheme can be unlimited without affecting its computational efficiency due to the trivial computing time required by the NN to perform one MCS. The difference in the computational time needed by the NN1 methodology, for different numbers of simulations, compared to that needed by NN2 is due to the fact that in the first methodology the computational time for the generation of the training set depends on the number of MC simulations which are required for checking the feasibility with respect to the probabilistic constraint. For this example it is observed that when the number of required simulations is less than 500, the NN1 methodology outperforms the NN2, the two methodologies perform similarly for 1000 simulations, while in the case of more than 5000 simulations the NN2 needs one order of magnitude less computational time compared to both conventional and NN1 RBDO methodologies. The computational time for the conventional RBDO is decreased by 70% when the NN1 methodology is employed.

8.2. 39-bar truss—RDO test example

A three-dimensional 39-bar truss shown in Fig. 3 is considered for testing the efficiency of the proposed RDO methodology. The height of the structure is 16 m (Fig. 3(b)), while its basis is an equilateral triangle of side 6.93 m (Fig. 3(c)). Two objective functions are used: the weight and the standard deviation of a characteristic nodal displacement representing the response of the structure; in this test example the variance of the top displacement in the x direction is selected. The design variables considered are the dimensions of the members of the structure, four groups in total, taken from the Circular Hollow Section (CHS) table of the Eurocode. For each design variable, two stochastic

variables are assigned: the external diameter d and the thickness t of the circular hollow section. A vertical load $V = 2$ kN is applied to all nodes, while a probabilistic horizontal load F of mean value 8 kN is applied to the top nodes in the x direction.

For this test example, three types of constraint have been considered: (i) stress, (ii) compression force (for buckling) and (iii) displacement constraints, as imposed by the European design code [16]. The stress constraints considered can be written as follows:

$$\begin{aligned} \sigma_{\max} &\leq \sigma_a \\ \sigma_a &= \frac{\sigma_y}{1.10} \end{aligned} \quad (22)$$

where $\sigma_{\max}$ is the maximum axial stress in each element group for all loading cases, σ_a is the allowable axial stress according to Eurocode 3 [16] and σ_y is the yield stress. For members under compression an additional buckling constraint is implemented:

$$\begin{aligned} |P_{c,\max}| &\leq P_{cc} \\ P_{cc} &= \frac{P_e}{1.05} \\ P_e &= \frac{\pi^2 EI}{L_{\text{eff}}^2} \end{aligned} \quad (23)$$

where $P_{c,\max}$ is the maximum axial compression force for all loading cases, P_e is the critical Euler buckling force in compression, taken as the first buckling mode of a pin-connected member, and L_{eff} is the effective length. The effective length is taken equal to the actual length of the member. Similarly, the displacement constraints can be written as

$$|d| \leq d_a \quad (24)$$

where d_a is the limit value of the displacement at a certain node or the maximum nodal displacement, taken as 200 mm.

The type of probability density function, the mean value and the variance of the random parameters are shown in

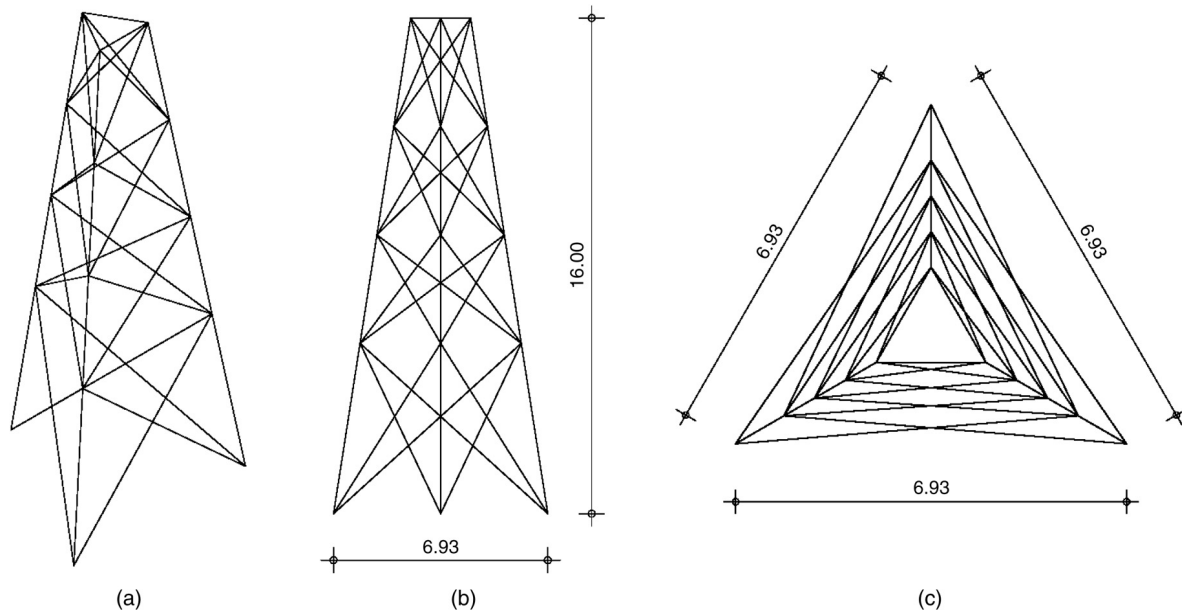


Fig. 3. Three-dimensional 39-bar truss example: (a) 3D view, (b) side view, (c) top view.

Table 3

Three-dimensional 39-bar truss example: characteristics of the random variables

		Probability density function	Mean value μ	Standard deviation σ	σ/μ (%)	95% of values within
E (kN/m ²)	Young's modulus	Normal	2.10E+08	1.50E+07	7.14	(1.81E+08, 2.39E+08)
σ_y (kN/m ²)	Allowable stress	Normal	355 000	35 500	10.00	(2.85E+05, 4.25E+05)
F (kN)	Horizontal loading	Normal	8	3	37.50	(2.12, 13.88)
d	CHS diameter	Normal	d_i^a	$0.02d_i$	2	$(0.9608d_i, 1.0392d_i)$
t	CHS thickness	Normal	t_i^a	$0.02t_i$	2	$(0.9608t_i, 1.0392t_i)$

^a Taken from the Circular Hollow Section (CHS) table of the Eurocode, for every design.

Table 3. For this test case the $(\mu + \lambda)$ -ES approach is used with $\mu = \lambda = 5$, while a sample size of 1000 simulations is taken for the MCS. The resultant Pareto front curve is depicted in Fig. 4, with the weight of the structure and the standard deviation of the horizontal displacement on the horizontal and vertical axis, respectively. The Pareto front curve shows a strong conflict between the two objective functions in question.

9. Conclusions

In most usual cases the optimum design of structures is based on deterministic parameters and is focused on the satisfaction of the associated deterministic constraints. So far many papers have been devoted to this research field and efficient methods have been proposed. The deterministic optimum cannot be considered as a realistic optimum design since in real life a number of random factors affect the design, the manufacturing and the life cycle of a structure. In order to find a more realistic optimum the designer has to take into account the variation of the random parameters.

The aim of the proposed RBDO procedure was threefold: to reach an optimized design with controlled safety margins

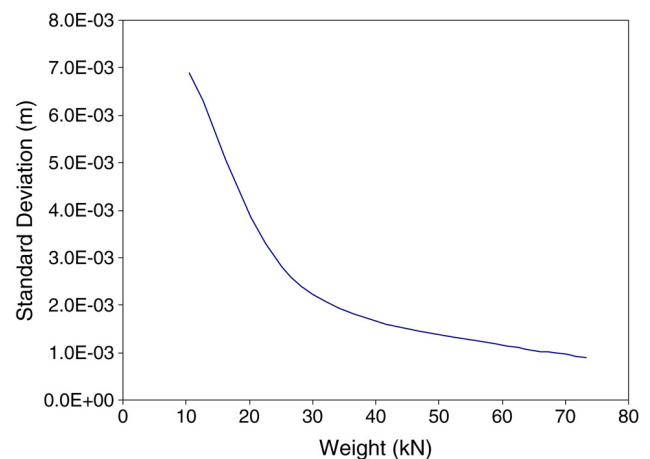


Fig. 4. Three-dimensional 39-bar truss example: the Pareto front curve for the RDO problem.

with regard to various model uncertainties, while at the same time to minimize the weight of the structure and to reduce substantially the required computational effort. The solution of realistic RBDO problems in structural mechanics is an extremely computationally intensive task. In the test examples

considered, the conventional RBDO procedure was found to be almost two orders of magnitude more computationally expensive than the corresponding deterministic optimization procedure. The aim of decreasing the computational cost by at least one order of magnitude was achieved using: (i) NN predictions to perform both deterministic and probabilistic constraints checks; or (ii) NN predictions for performing the structural analyses involved in MCS.

Furthermore, in order to account for the response of the structure that is affected by the randomness of structural parameters, an RDO formulation of the optimization problem has to be used, as shown in this work. The two ends of the Pareto front curve represent two designs, one for which the weight of the structure is the dominant criterion, that corresponds to the deterministic-based optimum, and the other for which the variance of the response is considered as the governing criterion. The intermediate Pareto optimal solutions are compromise solutions, between the two conflicting criteria. The deterministic-based formulation leads to an optimum design having the minimum weight, yet the maximum variance of the response. This wide variation of the response might cause the violation of some constraints that could consequently put the quality of the final design in doubt. The proposed two-stage Evolution Strategies method for treating multi-objective robust design optimization problems proved to be an efficient and reliable optimization tool, in generating the Pareto front curve.

References

- [1] Schueller GI, editor. Computational stochastic structural mechanics and analysis as well as structural reliability. *Comput Methods Appl Mech Engrg* 2005 [special issue].
- [2] Allen M, Maute K. Reliability-based shape optimization of structures undergoing fluid-structure interaction phenomena. *Comput Methods Appl Mech Engrg* 2005.
- [3] Moses F. Problems and prospects of reliability-based optimization. *Eng Struct* 1997;19(4):293–301.
- [4] Kong JS, Frangopol D. Life-cycle reliability-based maintenance cost optimization of deteriorating structures with emphasis on bridges. *Struct Eng* 2003;129(6):818–28.
- [5] Lee K-H, Park G-J. Robust optimization considering tolerances of design variables. *Comput Struct* 2001;79:77–86.
- [6] Messac A, Ismail-Yahaya A. Multiobjective robust design using physical programming. *Struct Multidisc Optim* 2002;23:357–71.
- [7] Doltsinis I, Kang Z. Robust design of structures using optimization methods. *Comput Methods Appl Mech Engrg* 2004;193(23–26):2221–37.
- [8] Tsompanakis Y, Papadrakakis M. Large-scale reliability based structural optimization. *Struct Multidisc Optim* 2004;26(6):429–40.
- [9] Pareto V. *Cours d' économie politique*, vol. 1–2. Rouge, Lausanne, 1896–1897.
- [10] Papadrakakis M, Lagaros ND, Plevris V. Multi-objective optimization of space structures under static and seismic loading conditions. *Eng Optim* 2002;34:645–69.
- [11] Hurtado JE, Alvarez DA. Neural-network-based reliability analysis: a comparative study. *Comput Methods Appl Mech Engrg* 2001;191(1–2):113–32.
- [12] Papadrakakis M, Papadopoulos V, Lagaros ND. Structural reliability analysis of elastic–plastic structures using neural networks and Monte Carlo simulation. *Comput Methods Appl Mech Engrg* 1996;136:145–63.
- [13] Lagaros ND, Papadrakakis M. Improving the condition of the Jacobian in neural network training. *Adv Eng Softw* 2004;35(1):9–25.
- [14] Zitzler E, Deb K, Thiele L. Comparison of multiobjective evolutionary algorithms: empirical results. *Evol Comp* 2000;8(2):173–95.
- [15] Papadimitriou C. Pareto optimal sensor locations for structural identification. *Comput Methods Appl Mech Engrg* 2005;194:1655–73.
- [16] Eurocode 3. Design of steel structures, Part 1.1: General rules for buildings. CEN, ENV 1993-1-1/1992.