First South-East European Conference on Computational Mechanics, SEECCM-06
(M. Kojić, M. Papadrakakis (Eds.))
June 28-30, 2006, Kragujevac, Serbia
University of Kragujevac

Metamodel Assisted Techniques for Structural Optimization

V. Plevris[1], N. D. Lagaros[1], D. Charmpis[2], M. Papadrakakis[1]

[1] Institute of Structural Analysis & Seismic Research, School of Civil Engineering,
National Technical University, Zografou Campus, Athens 15780, Greece
e-mail: {vplevris, nlagaros, mpapadra}@central.ntua.gr
[2] Department of Civil & Environmental Engineering,
University of Cyprus, 75 Kallipoleos Str., P.O. Box 20537, 1678, Nicosia, Cyprus
e-mail: charmpis@ucy.ac.cy

Abstract

In this work the application of Evolution Strategies (ES) combined with Neural Networks (NN) is investigated, in various structural optimization problems, in an effort to increase the robustness as well as the computational efficiency of the optimization procedure. The use of NN is motivated by the time-consuming repeated Finite Element analyses required for ES during the optimization process. The suitability of NN predictions is investigated in a number of structural problems optimized using ES and the computational advantages of the proposed methodologies are demonstrated. In addition, a thorough investigation is performed on the selection of the training schemes used for the NN learning procedure in order to ensure the generality and robustness of the proposed methodologies.

For each problem a NN is trained utilizing information generated from a number of properly selected FE analyses. The data from the analyses are processed in order to obtain the necessary input and output pairs which are subsequently used to produce a trained NN. The trained NN is then used in order to predict the response of the structure in terms of objective and constraints function values due to the different sets of design variables. It appears that the use of a properly selected and trained NN can eliminate any limitation on the dimensionality of the problem, due to a drastic reduction of the computing time required for the repeated FE analyses.

Key words: Structural optimization, Evolution Strategies, Neural Networks.

1. Introduction

Evolutionary Algorithms (EA), like Genetic Algorithms (GA) or Evolution Strategies (ES), are widely accepted today as a family of effective methods for handling large-scale structural optimization problems and have been successfully applied to a variety of applications in Computational Structural Mechanics [11,19,21]. EA are capable of locating (near-) optimum solutions within large and irregular search spaces by maintaining a population of potential solutions in the context of an evolution-based procedure. However, locating optimal structural designs using EA is a task with high computational cost, since a complete Finite Element (FE) analysis needs to be carried out for each parent and offspring design vector of the populations considered.

On the other hand, artificial intelligence or soft computing techniques have emerged over the last ten years as a valuable tool used to replace time consuming computational tasks in many scientific and engineering applications. The use of such techniques, like Neural Networks (NN), to predict FE analysis results has been previously studied in the context of optimal design of structural systems [1,13], as well as in some other areas of structural engineering applications, such as structural damage assessment, structural reliability analysis, FE mesh generation and fracture mechanics [16]. The main concern in substituting FE computations by NN-based schemes has been to determine properly trained NN configurations and mechanisms to detect and correct false NN predictions, in order to ensure the reliability and accuracy of the NN results.

A NN can provide computationally inexpensive estimates of Finite Element analysis outputs required during the optimization process. A trained neural network presents some distinct advantages. It provides a rapid mapping of a given input into the desired output quantities, thereby enhancing the efficiency of the structural analysis process. This major advantage of a trained NN over the conventional procedure, under the provision that the predicted results fall within acceptable tolerances, leads to results that can be produced in a few clock cycles, representing orders of magnitude less computational effort than the conventional computational process. The learning algorithm, which was employed for the NN training of the present study, is the well-known Back Propagation (BP) algorithm.

The aim of the present study is to illustrate the benefits of the application of neural networks in various Structural Optimization problems. The paper is divided into two parts:

In the first part, the design optimization of skeletal structures is studied by implementing an ES-based procedure. Each parent and offspring design of the populations involved in the ES process requires a constraints check, which facilitates decision making regarding the feasibility or infeasibility of the structural design considered. Thus, a FE solution is performed for each design, in order to evaluate the displacement and stress constraints specified for the structural problem at hand. The aim of the NN is to reliably predict the feasibility or infeasibility of structural designs and therefore avoid computationally expensive FE analyses in the framework of the ES optimization procedure. The proposed NN implementation is adaptive in the sense that the utilized NN configuration is appropriately updated as the ES process evolves by performing NN retrainings using information gradually accumulated during ES optimization steps. Thus, this NN scheme is capable of adapting to a particular structural optimization run by gradually acquiring prediction capabilities for the regions of the overall design space that are actually visited by the ES procedure. The prediction capabilities and the computational advantages offered by the proposed adaptive NN strategy in the context of ES-based structural design optimization are investigated on both sequential and parallel computing environments.

In the second part, a NN is employed in order to predict the collapse load for different values of the basic random variables in a large-scale Reliability-based optimization problem. The calculation of the collapse load is used by the Monte Carlo Simulation (MCS) method for reliability analysis, incorporating the importance sampling technique for the reduction of the sample size. The results of the reliability analyses are used to verify the feasibility or not of the design with respect to the probabilistic constraint functions. This is achieved with a proper training of the NN. The NN training comprises the following tasks: (i) select the proper training set, (ii) find a suitable network architecture and (iii) determine the appropriate values of characteristic parameters such as the learning rate and momentum term. The basic NN configuration employed in the study is selected to have one hidden layer, while the optimization part is performed with evolution strategies. The elasto-plastic analysis phase, required by the MCS, is replaced by a neural network predictor in order to predict the necessary data for the MCS procedure. The use of neural networks is motivated by the approximate concepts inherent in reliability analysis and the time consuming repeated analyses required by the MCS. A training algorithm is implemented for training the NN utilizing available information generated from selected elasto-plastic analyses.

2. Part I: Structural optimization combining ES and NN

This first part is focused on accelerating the optimization process with the use of NN-based schemes for the prediction of the constraints checks results. A neural network attempts to create a desired relation for an input/output (I/O) set of *m* learning patterns. This set, which is called training set, consists of a finite number of *m* pairs $(\mathrm{inp},\mathrm{tar})\in \mathrm{R}^k\times\mathrm{R}$. 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 $\mathrm{R}^k\rightarrow\mathrm{R}$ between these two spaces is the back propagation algorithm. This algorithm tries to determine a set of parameters (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. Based on previous study [14], the Levenberg-Marquard method [10] is used in the present study to minimize the residual values between calculated and desired results in the NN training procedure. Moreover, NN generalization is improved by determining in an automated fashion regularization parameters with the Bayesian framework by MacKay [12].

2.1 Conventional NN training

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 NN training, and finally the selection, training and testing of a suitable NN configuration. The second phase is the ES optimization process, but instead of conducting 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 selection of appropriate I/O training data is one of the most important factors in NN training. The number of training patterns used is not the only concern, since the distribution of samples may be of greater importance. Acceptable results can be expected by the trained NN scheme only if its training set includes data over the entire range of the design space. Thus, the selection of I/O training patterns 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, the training set can be chosen automatically based on a uniform distribution of the design variables in the design space. When a NN scheme is utilized to predict results of ES constraints checks, the I/O data required for each design contributing to the NN training set consists of: (a) the values of the design variables for the particular design (input data) and (b) the outcome of the constraints check (determined through FE analysis) denoting feasibility or infeasibility of the design (output data).

The combined ES-NN(M) methodology, where M is the size of the NN training set, is divided in the training phase of the NN and the ES-NN optimization. In this study the number of patterns used for NN training in the framework of the ES-NN(M) algorithm is M=200. This selection is based on a previous study by the authors [14].

2.2 Adaptive NN training

The training set for the conventional ES-NN scheme described in the previous subsection generally consists of I/O data corresponding to a limited number of structural designs, therefore the prediction capabilities offered cannot cover effectively the entire search space available. However, as the ES optimization process progresses, the designs produced after several generations become more and more different compared to the initial training data used for NN training at the beginning of the optimization procedure. Thus, a NN scheme trained according to an initially selected training set may become gradually incapable of yielding reliable predictions for constraints check results, since the entire design space is poorly represented by the limited number of randomly generated training patterns, leading to unreliable NN predictions.

Two undesirable cases may occur as a result of NN's incapability to correctly predict the feasibility or infeasibility of offspring designs: *A feasible design may be wrongly predicted as infeasible,* or *An infeasible design may be wrongly predicted as feasible.* This NN prediction claiming feasibility of the design will be checked with a standard FE analysis, which will reveal the wrong prediction made by the NN scheme. Thus, the design will finally not be involved in the optimization process, since it will be discarded based on computed (not predicted) information obtained through FE analysis.

To facilitate the discussion in the sequel of the present work, offspring structural designs are codified according to the NN predictions made for their feasibility or infeasibility. A NN prediction may be either Correct (*C*) or Wrong (*W*), while a NN configuration may predict either a Feasible (*F*) or an Infeasible (*I*) design. Hence, an offspring design can be characterized as either *CF*, *CI*, *WF* or *WI*, where the first letter of these abbreviations is associated with the correctness or not of the NN prediction (options: *C* or *W*), while the second letter expresses the NN output regarding feasibility or infeasibility of the design (options: *F* or *I*).

Some weak points of conventional NN-based ES procedures can be overcome with an adaptive NN strategy, which updates the utilized NN configuration as the ES process evolves. The proposed adaptive NN scheme performs NN retrainings using information gradually accumulated during ES execution and is therefore capable of adapting to a particular structural optimization run by gradually acquiring prediction capabilities for the regions of the overall design space that are actually visited by the ES procedure.

2.2.1 Definition of criterion for deciding NN retraining

The criterion, according to which the resorting to NN retraining is decided, must be allowed to show some tolerance with wrong NN predictions, in order to avoid wasting computing time due to unnecessarily large numbers of NN retrainings. Thus, the detection of a single or a few wrong NN predictions should not immediately invoke the NN retraining procedure, but rather act as an alarm sign that the current NN configuration may be loosing its prediction reliability. In the present work NN retraining is requested only after detecting 10 wrong NN predictions corresponding to *WF*-designs or 5 consecutive wrong NN predictions. This relaxed criterion shows some tolerance in the mistakes made by the current NN configuration on the anticipation that the correct NN predictions in forthcoming constraints checks will be substantially more than the wrong ones.

2.2.2 Composition of NN retraining set

The straightforward solution to the second issue posed regarding the composition of the NN retraining set is to add to the previous training set the I/O data of the 10 *WF*-designs, which have contributed to the fulfilment of the criterion for NN retraining. This retraining approach is based on the rationale that the current NN configuration should learn from its mistakes, in order to avoid repeating them again. However, a retraining oriented towards healing specific NN prediction weaknesses may confuse the utilized NN scheme, since NN prediction reliability may be enhanced in some regions of the design space and disturbed elsewhere. Therefore, it may be more effective to enrich the NN retraining set with I/O data belonging not only to *WF*-designs.

Adaptive NN training scheme	Number of designs contributing I/O data to the NN retraining set			
	WF-designs	*CF*-designs	*WI*-designs	Total
A	10	-	-	10
B	10	2	-	12
C	10	4	-	14
D	10	2	0-2	12-14
E	10	2	0-4	12-16
F	10	4	0-2	14-16
G	10	4	0-4	14-18

Table 1. Definition of adaptive NN training schemes

Seven NN retraining schemes are examined in this study. They are summarized in Table 1 and explained in the sequel of this sub-subsection. The idea employed by these schemes is to enrich the NN training set with I/O data corresponding to a relatively small number of wrong or nearly wrong NN predictions and retrain the NN by initiating the training with the weights coefficients of the previous NN configuration. To explain the meaning of nearly wrong NN predictions, it should be mentioned first that the NN implementation used in the present work employs a sigmoid transfer function, whose output values are real numbers in the interval [0,1]. A NN output in the range [0,0.5] nominates a design as feasible, while a NN output within (0.5,1] nominates a design as infeasible, i.e. the border value of 0.5 distinguishes predicted feasibility and infeasibility. A nearly wrong NN prediction about a *CF*-design is a prediction, which nominates the design as feasible (i.e. the prediction is based on a NN output in the interval [0,0.5]) and is confirmed regarding its correctness through FE analysis results, but the associated NN output is close to the border value of 0.5, which would yield a wrong NN prediction). Hence, the first three NN retraining schemes examined are: (i) the **A**-scheme, in which NN retraining is performed by adding to the training set the 10 extra training patterns of detected *WF*-designs; (ii) the **B**-scheme, which adds to the training set I/O data of the 10 *WF*-designs plus 2 *CF*-designs with nearly wrong NN predictions; (iii) the **C**-scheme, which exploits additional training patterns of 10 *WF*-designs plus 4 *CF*-designs with nearly wrong NN predictions.

If the initial NN training set is small (e.g. consists of only 6 patterns, as in certain runs of the present work), then the retraining schemes **A**, **B** and **C**, which take into account only *WF*-designs and maybe some *CF*-designs with nearly wrong NN predictions, will force the retrained NN configuration to nominate almost every design considered as an infeasible one. This observation has led to the investigation of four additional retraining schemes making use of I/O data belonging also to *WI*-designs. As a general rule, however, all designs contributing I/O data to the NN training set should have their associated NN predictions regarding feasibility or infeasibility checked through FE analysis. As already stated, *WF* and *CF*-designs are detected using FE results, but *WI*-designs are generally not detectable during the ES process (NN outputs predicting infeasibility are accepted without verification through FE analysis). Thus, in order to detect *WI*-designs for the four additional retraining schemes, a number of NN predictions nominating designs as infeasible have to be checked through FE analysis. Such checks yield the desired *WI*-designs, but reveal also *CI*-designs, which are ignored and not included in the retraining set.

Following the above discussion, the four additional NN adaptive training schemes are defined as follows: (iv) the **D**-scheme adds to the training set extra training patterns of 10 *WF*-designs plus 2 *CF*-designs with nearly wrong NN predictions plus a number of *WI*-designs, which are detected among 2 randomly chosen designs predicted as infeasible (i.e. the *WI*-designs detected among the 2 randomly chosen designs predicted as infeasible are added to the training set, while the remaining *CI*-designs are ignored); (v) the **E**-scheme takes into account I/O data of 10 *WF*-designs plus 2 *CF*-designs with nearly wrong NN predictions plus a number of *WI*-designs, which are detected among 4 randomly chosen designs predicted as infeasible; (vi) the **F**-scheme enriches the retraining set with I/O data of 10 *WF*-designs plus 4 *CF*-designs with nearly wrong NN predictions plus a number of *WI*-designs, which are detected among 2 randomly chosen designs predicted as infeasible; (vii) the **G**-scheme adds training patterns due to 10 *WF*-designs plus 4 *CF*-designs with nearly wrong NN predictions plus a number of *WI*-designs, which are detected among 4 randomly chosen designs predicted as infeasible.

2.2.3 ES-NNX algorithms

Two algorithms are proposed for implementing the seven adaptive NN training schemes A-G in the framework of ES optimization. The first algorithm denoted as **ES-NNX(M)**, where **X** is the adaptive training scheme adopted and M is the size of the starting NN training set,

In this study the number of patterns used for initial NN training in the framework of the ES-NNX(M) algorithm is M=50 or M=100. These selections correspond respectively to 25% or 50% of the training set size used in the conventional ES-NN(M) methodology (see section 4.1).

The second algorithm denoted as **ES-NNX(μ)**, where **X** is the adaptive training scheme adopted and μ is the size of the starting NN training set coinciding with the number of ES parent designs.

3. Part II: The application of NN in Reliability-based Structural Optimization

In sizing optimization problems the aim is to minimize the weight 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 and to ensure that the probability of failure of the structure is within acceptable limits. The probabilistic constraints enforce the condition that the probability of a local or a system failure is smaller than a certain value (i.e. 10^{-3}). In this work the overall probability of failure of the structure, as a result of a limit elasto-plastic analysis, is taken as the global reliability constraint.

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

The optimization part is solved using evolution strategies (ES), which in most cases are more robust and present a better global behaviour than mathematical programming methods (Papadrakakis et. al. [17]). The limit state elasto-plastic analyses required during the MCS are replaced by NN predictions. The use of NN is motivated by the approximate concepts inherent in reliability analysis and the time consuming repeated analyses required for MCS. An NN is trained first 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 the NN training. The trained NN is then used to predict the critical load factor due to different sets of basic random variables. It appears that the use of a properly selected and trained NN can eliminate any limitation on the sample size used for MCS and on the dimensionality of the problem, due to the drastic reduction of the computing time required for the repeated limit state elasto-plastic analyses.

3.1 The Monte Carlo Simulation method

In reliability analysis the MCS method is often employed when the 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. Although 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, the computational effort involved in conventional MCS is excessive. For this reason a lot of 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.

3.2 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 [3]. The key-idea of this technique is to obtain a non-negative sampling density located in the neighbourhood of the most probable failure point. The selection of an appropriate important sampling density function $g_x(x)$ is of critical importance for both the efficiency and the accuracy of the MCS. A successful choice of $g_x(x)$ yields reliable results and reduces significantly the number of simulations, while a misleading choice may produce inaccurate results.

3.3 Reliability-based structural optimization using MCS, ES and NN

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

At each ES cycle (generation) a number of MCS are carried out. In order to replace the time consuming limit elasto-plastic analyses needed by MCS for each design, a training procedure is performed based on the data collected from M conventional limit elasto-plastic analyses. After the selection of the suitable NN architecture the training procedure is performed with M=30 data sets, in order to obtain the I/O pairs needed for the NN training. After the training phase is concluded the trained NN replaces the conventional limit elasto-plastic analyses, for the current design.

A few tens of limit elasto-plastic analyses have been found sufficient for the example considered to produce a satisfactory training of the NN. A fully connected network is used. The number of conventional step-by-step limit analysis calculations performed in order to built up the proper data for the training set is in the range of thirty. This selection is based on the requirement that the full range of possible results should be represented in the training procedure. For the application of the NN simulation and 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.

4. Test examples

4.1 Test example 1: Adaptive NN training in Structural Optimization

In this test example, a 3D bus frame has been considered, that is shown in Fig. 1, which consists of 1,269 elements, 753 nodes and 4,489 degrees of freedom in total. The cross sections of all frame members are assumed to be tubular. The members of the 3D bus involved in the optimization process are divided into 15 independent groups, which are illustrated in Figure 1 with different colors. Each of these groups corresponds to a design variable taking values from a database containing members with various cross-sectional geometric properties. The objective function to be minimized is the weight of the structure, while only stress constraints are imposed in the bus frame's elements. The bus structure is designed taking into account a combination of loading conditions including dead loads, breaking (mass×negative acceleration loading at the y-direction) and asymmetrical vertical loading (mass×acceleration loading at the z-direction).

The following abbreviations are used in this section: ES refers to the standard implementation of the ES optimization procedure, in which constraints checks are performed using conventional FE analyses without resorting to NN predictions; ES-NN(M) refers to the conventional ES combined with the NN methodology; ES-NNX(M) and ES-NNX(μ) refer to the algorithms combining ES with an adaptively trained NN (the adaptive NN training scheme adopted is designated by NNX).

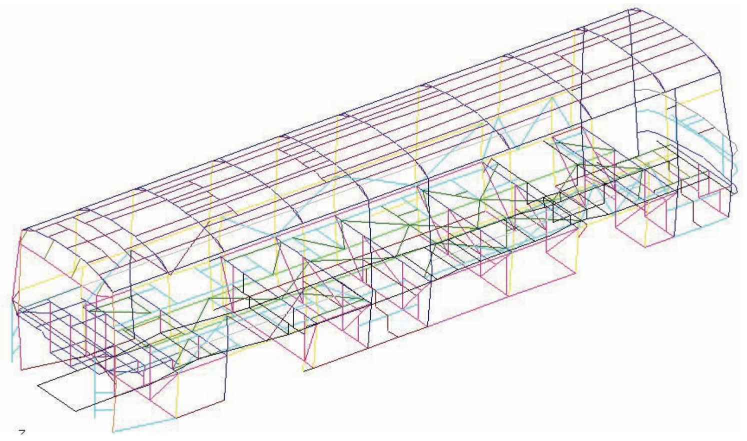

Fig. 1. The 3D bus frame example

The (6+12)-ES scheme was selected for this test example, as it showed the best performance compared to other ES schemes, in a parametric study that was carried out. A performance comparison of various adaptive retraining schemes versus the conventional non-adaptive one is shown in Table 2. According to these results, the non-adaptive scheme requires the largest number of training patterns (M=200). The adaptive NN configurations require substantially smaller numbers of training patterns in total. In particular, the A-scheme using μ=6 patterns as an initial training set results in 16 accumulated training patterns only and just 2 adaptive steps.

Starting training set	Adaptive NN training scheme	Final training set	Retraining steps	FE analyses
200	-	200	0	200
100	A	130	4	216
50	A	90	5	170
6	A	16	2	170
6	B	30	3	156
6	C	62	5	163
6	D	33	3	176
6	E	93	7	166
6	F	82	6	181
6	G	91	6	179

Table 2. Bus frame example – Performance comparison of adaptive NN training schemes

The performance of various ES methodologies in terms of both computing time required and optimum design achieved is presented in Table 3, where the computations are performed in a sequential computing environment using the direct solver. A first remark on the results shown is that the overall computing time consumed by the combined methodologies ES-NNX is about 60% to 80% less than the time required by standard ES. From Table 3 it can be seen that despite the use of a large number of patterns for NN training by the non-adaptive ES-NN(200) methodology the optimization algorithm has not managed to converge to a feasible design. Among all ES methodologies examined, ES-NNA(6) performs best in terms of computing time demands, but the small number of 16 training patterns used overall seems to be inadequate to lead the ES process to the optimum design achieved by other ES methodologies. Therefore, ES-NNE(6), ES-NNF(6) and ES-NNG(6) can be considered as the overall best performing methodologies for this test example, since they achieve the best quality design (15% less weight) at a 30% more computing time than ES-NNA(6). Table 3 also shows that training and retraining consume a fraction of computing time compared to the entire optimization process. As a general remark, it can be said that the best adaptive scheme corresponds to a compromise between the total training patterns used, the retraining steps required and the optimum design achieved.

Optimization methodology	Time (s)				Weight (kN) of optimum design
	1st Training	Retrainings	FE Analyses - Optimizer	Total	
ES	-	-	877	877	21.4
ES-NN(200)	192	-	230	422	19.5 *
ES-NNA(100)	138	22	181	341	23.1
ES-NNA(50)	117	30	181	328	23.1
ES-NNA(6)	4	7	166	177	25.1
ES-NNB(6)	4	15	174	193	25.1
ES-NNC(6)	4	31	188	223	23.1
ES-NND(6)	4	16	177	197	25.1
ES-NNE(6)	4	45	193	242	21.4
ES-NNF(6)	4	40	191	235	21.4
ES-NNG(6)	4	38	191	233	21.4

* infeasible design

Table 3. Bus frame example - Performance comparison of optimization methodologies

4.2 Test example 2: NN in Reliability-based Structural Optimization

The model of the second test example consists of 63 elements with 180 degrees of freedom as shown in Figure 2. The length of the beams and the columns of the frame is L_1=7.32 m and L_2=3.66 m, respectively. The structure is loaded with a 19.16 kPa gravity load on all floor levels and a lateral load of 110 kN applied at each node in the front elevation along the z direction. The members of the structure are divided into five groups, as shown in Figure 2, each one having two design variables. The deterministic constraints are eleven, two for the stresses of each element group and one for the inter-storey drift. The type of probability density functions, mean values, and variances of the random parameters are presented in Table 4. The cross section of each member of the space frame considered is assumed to be a I-shape and for each member two design variables are allocated, the cross-sectional dimensions b, h, while the mean value is taken as the current value of the corresponding design variable s_i.

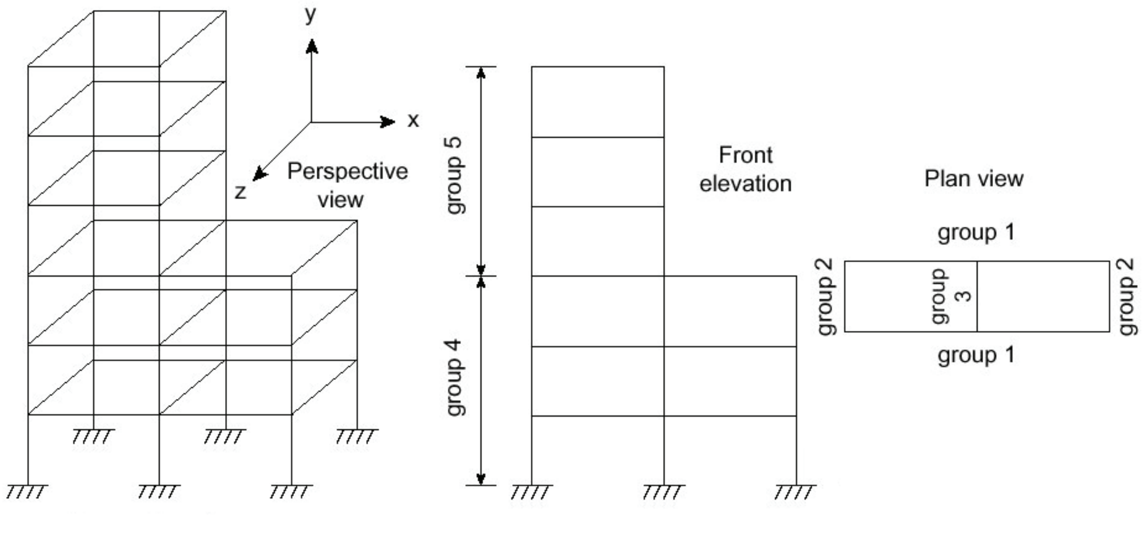

Fig. 2. Description of the six-storey frame

Random variable	Probability density function (pdf)	Mean value	Standard deviation (σ)
E	N	200	20
σ_y	N	25.0	2.5
b,h	N	s_i	$0.1s_i$
Loads	Log-N	640	20

Table 4. Characteristics of the random variables for the six-storey frame

The objective function of the problem 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 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 limited to 1.5% of the height of each storey.

The probabilistic constraint is imposed on the probability of structural collapse due to successive formation of plastic hinges and is set to p_a=0.001. The probability of failure caused by uncertainties related to material properties, member geometry and loads of the structure 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 (pdf), while random variables associated with material properties and cross-section dimensions follow a normal pdf. The required importance sampling function $g_x(x)$ for the loads is assumed to follow a normal distribution. The mean value of $g_x(x)$ corresponds to the failure load when all other random values are kept fixed to their mean values.

Optimization procedure	ES cycles	p_f	Optimum weight (tn)	Time (s)
DBO	41	0.166	67.5	177
RBO	79	0.001	77.8	54,126
RBO-NN	81	0.001	77.9	9,471

Table 5. Performance of the methods for the six-storey frame

For this test case the (μ+λ)-ES approach is used with μ=λ=5, while a sample size of 500 simulations is taken for the MCS combined with the Important Sampling technique. As it can be observed from Table 5 the optimum weight achieved by the RBO is 15% more than the deterministic one. On the other hand, the probability of failure for the deterministic optimum is inapplicable since it exceeds the accepted value of 10^{-3}. The proposed RBO-NN combination manages to achieve the optimum weight in one sixth of the CPU time required by the conventional RBO procedure.

3. Conclusions

An adaptive strategy for NN training can substantially improve the prediction reliability of a NN configuration in the context of the ES optimization procedure. The use of NN-based schemes to predict the outcome of constraints checks during the ES procedure can drastically accelerate the overall optimization process, since NN-predicted structural response was found to fall within acceptable tolerances.

A number of adaptive NN training schemes has been examined in the first part of this study. These schemes are based on two concepts using either a small or a large starting training set. The test examples examined in this study lead to the conclusion that a small initial training set performs better. This is explained by the fact that the proposed strategy of adaptively creating the NN training set is based on the idea of gradually providing a NN configuration with prediction capabilities for the regions of the overall design space that are actually visited by the ES procedure. This concept is best realized using a small initial training set, which is adaptively enriched with few but really effective NN training patterns. It has been observed that, when a small initial training set is used, it is necessary to gradually enhance the training set with additional training patterns corresponding to both feasible and infeasible designs, as implemented in adaptive training schemes D, E, F and G.

The proposed optimization methodology, which combines the ES algorithm with an adaptively trained NN predictor for constraints checks, can effectively handle computationally intensive optimization problems by implementing efficient domain decomposition solvers in both sequential and parallel computing environments. The numerical results reported in this study reveal the computational advantages offered by the proposed methodology over standard ES. More specifically, if we consider the conventional ES procedure equipped with a sequential direct solver as the reference optimization methodology, then for the test examples examined in this study the proposed algorithm combining ES and NN achieves the same optimum weight obtained by standard ES in about 4 to 5 times less computing time on a single computer and in 30 times less computing time using a parallel processing environment of 12 processors. To conclude, the ES algorithm combined with adaptive NN was found to provide high-quality results in affordable computing time, making this way the structural optimization process more tractable in engineering practice.

As it can be clearly seen from the second part of the study, the solution of realistic RBO problems in structural mechanics is an extremely computationally intensive task. In the test example considered the conventional RBO procedure was found over forty times more expensive than the corresponding deterministic optimization procedure. The aim of the proposed RBO procedure is to increase the safety margins of the optimized structures under various model uncertainties, while at the same time minimize the weight of the structure as well as the additional computational cost. This goal was achieved using NN predictions to perform the structural analyses involved in MCS.

References

[1] Adeli H., Karim A., Neural network model for optimization of cold-formed steel beams. J. Struct. Engrg. (ASCE), 123(11): 1535-1544, 1997.
[2] Bäck T., Schwefel H.P., An overview of evolutionary algorithms for parameter optimization. J. of Evol. Comp., 1(1): 1-23, 1993.
[3] Bucher C.G., Adaptive Sampling - An iterative Fast Monte Carlo Procedure, Structural Safety J., 5, 119-126, (1988).
[4] Charmpis D.C., Papadrakakis M., Subdomain cluster generation for domain decomposition methods using graph partitioning optimization. Engrg. Comp., 20(8): 932-963, 2003.
[5] Coello C.A.C., Theoretical and numerical constraint-handling techniques used with ecolutionary algorithms: a survey of the state of the art. Comp. Meth. Appl. Mech. Engrg., 191(11): 1245-1287, 2002.
[6] Eurocode 3, Design of steel structures, Part1.1: General rules for buildings, CEN, ENV 1993-1-1/1992.
[7] Farhat C., Pierson K., Lesoinne M., The second generation FETI methods and their application to the parallel solution of large-scale linear and geometrically non-linear structural analysis problems. Comp. Meth. Appl. Mech. Engrg., 184: 333-374, 2000.
[8] Farhat C., Roux F.X., Implicit parallel processing in structural mechanics. Comp. Mech. Adv., 2(1): 1-124, 1994.

[9] Frangopol D.M., Moses F., Reliability-based structural optimization, in Advances in design optimization, H. Adeli (Ed.), Chapman-Hall, 492-570, (1994).

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

[11] Lagaros N.D., Papadrakakis M., Kokossalakis G., Structural optimization using evolutionary algorithms. Comp. & Struct., 80(7-8): 571-587, 2002.

[12] MacKay D.J.C., A practical Bayesian framework for back-propagation networks, Neural Comp., 4(2): 448-472, 1992.

[13] Papadrakakis M., Lagaros N.D., Reliability-based structural optimization using neural networks and Monte Carlo simulation. Comp. Meth. Appl. Mech. Engrg., 191(32): 3491-3507, 2002.

[14] Papadrakakis M., Lagaros N.D., Soft computing methodologies for structural optimization, App. Soft Comp., 3(3): 283-300, 2003.

[15] Papadrakakis M., Lagaros N.D., Tsompanakis Y., Structural optimization using evolution strategies and neural networks, Comput. Methods Appl. Mech. Engrg., Vol. 156, pp 309-333, 1998.

[16] Papadrakakis M., Papadopoulos V., Lagaros N.D., Structural reliability analysis of elastic-plastic structures using Neural Networks and Monte Carlo simulation, Comp. Methods Appl. Mech. Engrg., 136, 145-163, 1996.

[17] Papadrakakis M., Tsompanakis Y., Lagaros N.D., Structural shape optimization using Evolution Strategies. Engrg. Opt. J., 31: 515-540, 1999.

[18] Rixen D.J., Farhat C., A simple and efficient extension of a class of substructure based preconditioners to heterogeneous structural mechanics problems. Int. J. Num. Meth. Engrg., 44: 489-516, 1999.

[19] Sarma K.C., Adeli H., Bilevel parallel genetic algorithms for optimization of large steel structures. Comp.-Aided Civil & Infr. Engrg., 16(5): 295-304, 2001.

[20] Schueller G.I., Structural reliability – Recent advances, In the 7th International Conference on Structural Safety and Reliability (ICOSSAR '97), Kyoto, Japan, 1997.

[21] Soremekun G., Gurdal Z., Haftka R.T., Watson L.T., Composite laminate design optimization by genetic algorithm with generalized elitist selection. Comp. & Struct., 79(2): 131-144, 2001.

[22] Zhang L., Subbarayan G., An evaluation of back-propagation neural networks for the optimal design of structural systems: Part II. Numerical evaluation. Comp. Meth. Appl. Mech. Engrg., 191: 2887-2904, 2002.