MecSol2013
International Symposium on Solid Mechanics
April 18-19, 2013
Porto Alegre, Rio Grande do Sul, Brazil

# Optimization of a Transmission Line Tower under Random Wind Loads Considering Expected Consequences of Failure

**Wellison J.S. Gomes**
Department of Structural Engineering
São Carlos School of Engineering, University of São Paulo
wellison63@hotmail.com

**André T. Beck**
Department of Structural Engineering
São Carlos School of Engineering, University of São Paulo
atbeck@sc.usp.br

## ABSTRACT

Optimization of real structural systems considering uncertainties is a demanding task, as large FE models have to be solved thousands or millions of times. As a result, most applications found in the literature either: a. consider only small FE models; b. disregard nonlinearities in structural response; c. include limited number of design variables; d. solve only for local instead of global minima or e. disregard the expected consequences of failure. This last simplification is less obvious, but it allows known shortcuts to be adopted, reducing significantly the computational burden. The investigation reported in this paper advances simultaneously in areas b. to e. above. The article discusses the challenges in solving real structural optimization problems, taking into account the expected consequences of failure, and looking for the global optimum. The positional finite element method is used to obtain robust nonlinear structural responses, considering large displacements and material yielding. Structural reliability analysis is used to quantify probabilities of failure, given the inherent uncertainties in loads and resistance of structural materials. A hybrid BFGS-PSO algorithm is employed to efficiently solve the resulting global optimization problem. Artificial Neural Networks are used as surrogates for the objective function, significantly reducing the computational burden, but with no compromises in the quality of results. A challenging example is presented, involving the optimization of a steel-frame power-line transmission tower subject to random wind loads.

**Keywords:** structural optimization; optimum design; reliability analysis; risk optimization.

## 1 INTRODUCTION

Optimization of real structural engineering systems is a demanding task. Modelling the structural behaviour of real structures requires numerical models (e.g., FEM) of many degrees of freedom, which are computationally expensive to evaluate. Optimizing such structures requires hundreds to thousands of structural response evaluations. And the resulting optimal structures must be robust with respect to the uncertainties inherently present in loads and in the strength of structural materials.

In a competitive environment, structural systems have to be designed taking into account not just their functionality, but also expected construction and operational costs, and their capacity to generate profits. This capacity can be adversely affected by the costs of failure. Expected costs of failure quantitatively represent the different risks that construction and operation of a given facility

imply to the owner, to users, to employees, to the general public and/or to the environment. Uncertainty implies risk, and the possibility of undesirable structural responses.

In monetary terms, risk (or the expected cost of failure) is given by the product of failure costs by failure probabilities. Failure probabilities and failure consequences can be directly affected by structural design.

In structural engineering, economy and safety are generally considered to be competing goals. To the conventional structural engineer, increasing safety implies greater costs, and reducing costs may compromise safety. Hence, designing structural systems would involve a trade-off between safety and economy. In common engineering practice, this trade-off is addressed subjectively. When using structural design codes, the trade-off has already been decided by a code committee, which specifies safety coefficients to be used in design, and basic safety measures to be adopted in construction and operation. In deterministic structural optimization, this trade-off is completely neglected, because failure probabilities are not quantified. In classical [1-8] Reliability-Based Design Optimization (RBDO) the trade-off between safety and economy is also not addressed, because failure probabilities are constraints and not design variables. Robust design optimization [9, 10] searches for designs which are less sensitive to the existing uncertainties, but safety-economy tradeoffs are also not addressed.

However, when expected costs of failure are included in the design equation [11-21], one realizes that economy and safety are not competing goals. Safety is just another design variable which directly affects expected costs of failure. Since failure probabilities and consequences of failure are directly affected by structural design, optimum (minimum cost) design can only be achieved by quantifying uncertainties, probabilities of failure and costs of failure. In other words, optimum (minimum cost) design can only be achieved by quantifying expected costs of failure, and by treating safety as a design variable. This is called structural risk optimization herein and in a few other references [19- 21].

Optimization of real structural engineering systems is a demanding task... Even more demanding is the optimization of real structural engineering systems in consideration of the several sources of uncertainty which may affect system performance. Quantifying failure probabilities due to these uncertainties involves structural reliability analyses, which require repetitive solutions of the "deterministic" numerical FE models. When Monte Carlo simulation is used, these may reach thousands to millions for a single reliability analysis. Special subset simulation schemes have been devised for solving optimization problems under uncertainty, but these are only effective for reduced numbers of design variables. When the efficient FORM method is used for reliability analysis, still hundreds to thousands structural responses may be needed for each reliability analysis. When FORM is used, a nested optimization problem is obtained [1-8]. For classical RBDO problems, where failure probabilities are constraints and not design variables, a number of approaches have been proposed to avoid the nested optimization loops [1-8]. However, these shortcuts do not apply to risk optimization problems.

The authors are not aware of any similar shortcuts to solving structural risk optimization problems. Hence, each trial in a risk optimization solution requires at least one complete reliability analysis, which represents hundreds to thousands of structural response evaluations.

Moreover, it was found [20] that risk optimization problems possess many local minima. Hence, local optimization algorithms can at best improve a given initial design. Finding the (global) optimum design demands global optimization algorithms. This significantly increases the difficulties, as global optimization algorithms require many times more objective function evaluations than local optimization algorithms. Each objective function evaluation in risk optimization leads to a complete reliability analysis, which requires many structural (mechanical) response evaluations. Hence, the increase in computational cost is compounding. In this article, a special hybrid BFGS-PSO algorithm [22] is used to solve the global optimization problem. As

direct solution of this problem is very expensive, Artificial Neural Networks (ANN) are used as surrogate models for the objective function, in order to reduce the computational burden.

Another compounding difficulty arises when using global optimization algorithms to solve structural optimization problems. Local algorithms search for a better solution in a vicinity of a given initial design, which is normally a feasible and well-behaved design. Global optimization algorithms, on the other hand, have to test designs that are scattered all over the design space. Hence, it is easy to arrive at weird structural configurations, which would make no sense to a structural designer, but which end up being tested by the optimization algorithm. Weird designs can even belong to the failure domain, that is, they are more likely to fail than not. These weird designs can lead to numerical instabilities and convergence difficulties for both the non-linear (FE) mechanical models and for the reliability analysis algorithms.

In the present article, the positional finite element method [23-28] is used to compute non-linear structural responses. The positional FE method is a robust numerical analysis method, as it allows computing large displacements under material non-linearities. In the positional FEM, the displaced configuration is the primary unknown: displacements and rotations are evaluated afterwards. Equilibrium equations are evaluated in the displaced configuration. Material points are located by configuration-change functions and their gradients. Non-linear Cauchy-Green deformation measures are used, as well as their energy conjugate. Second-order Piola-Kirchhoff stress tensors are considered. With these stress-strain measures, the Saint Venant-Kirchhoff constitutive tensor is obtained. Stationarity of the total potential energy is used to arrive at the equilibrium equations. The Newton-Raphson method is used to solve the non-linear equations, with a consistent tangent stiffness matrix.

The article is laid out as follows. In Section 2, the risk optimization problem is formulated. Section 3 describes two different Artificial Neural Networks, which are used as surrogates to aid the solution of global risk optimization problems, as described in Section 4. Section 5 presents results for an application problem, involving the optimization of a power-line tower subject to random wind loads. Concluding remarks are presented in Section 6.

## 2 THE FORMULATION: RISK OPTIMIZATION PROBLEM

### 2.1 Structural reliability

Let **X** and **d** be vectors of structural system parameters. Vector **X** represents all random or uncertain system parameters, and includes geometric characteristics, resistance properties of materials or structural members, and loads. Some of these parameters are random in nature; others cannot be quantified deterministically due to uncertainty. Vector **d** contains all deterministic design parameters, like nominal member dimensions, partial safety factors, design life, parameters of inspection and maintenance programs, etc. Vector **d** may also include some parameters of random variables in **X**; for instance, the mean of a random variable may be a design parameter.

The existence of uncertainty implies risk, that is, the possibility of undesirable structural responses. The boundary between desirable and undesirable structural responses is given by limit state functions $g(\mathbf{X},\mathbf{d})$, such that:

$$\begin{aligned} \Omega_{\mathrm{f}}(\mathbf{d}) &= \{\mathbf{x} \mid g(\mathbf{X},\mathbf{d}) \le 0\} \\ \Omega_{\mathrm{s}}(\mathbf{d}) &= \{\mathbf{x} \mid g(\mathbf{X},\mathbf{d}) > 0\} \end{aligned} \tag{1}$$

Each limit state describes one possible failure mode of the structure, either in terms of serviceability or ultimate capacity. The probability of undesirable structural response, or probability of failure, is given by:

$$P_f\left(\mathbf{d}\right) = P\left[\mathbf{X} \in \Omega_f\left(\mathbf{d}\right)\right] = \int_{\Omega_f(\mathbf{d})} f_{\mathbf{X}}\left(\mathbf{x}\right) d\mathbf{x} \tag{2}$$

where $f_{\mathbf{X}}(\mathbf{x})$ is the joint probability density function of random vector $\mathbf{X}$. The probabilities of failure for individual limit states and for system failure can be evaluated using traditional structural reliability methods such as FORM, SORM and Monte Carlo simulation.

In the risk optimization problem, reliability analyses have to be repeated thousands of times. Hence, the algorithm used for reliability analysis has to be very efficient. In this paper, the First Order Reliability Method (FORM) is used, which is reasonably accurate and quite efficient. Importantly, the efficiency of FORM is equivalent for structural configurations leading to large or to very small failure probabilities. These different structural configurations have to be tested in global risk optimization.

In the FORM method, Eq. (2) is solved by means of a mapping from the space of the original random vector $\mathbf{X}$ to the so-called standard Gaussian space:

$$\mathbf{u} = T\left[\mathbf{x}\right]; \quad g_{\mathbf{u}}\left(\mathbf{u},\mathbf{d}\right) = g(T^{-1}[\mathbf{u}],\mathbf{d}) \tag{3}$$

where all components of vector $\mathbf{u}$ are independent and identically distributed standard Gaussian random variables. This mapping can be accomplished by means of the Nataf or Rosenblatt transformations. In the standard Gaussian space, the joint probability density $f_{\mathbf{U}}(\mathbf{u})$ is rotationally symmetric: hence, the point $\mathbf{u}^*$ on the limit state function $g_{\mathbf{u}}(\mathbf{u},\mathbf{d})=0$ which is closest to the origin represents the most probable failure point, also known as the design point. Hence, the design point search can be cast as a constrained optimization problem:

$$\begin{aligned} &\mathbf{u}^* = \arg\min \|\mathbf{u}\| \\ &\text{subject to } g_{\mathbf{u}}\left(\mathbf{u},\mathbf{d}\right) = 0 \end{aligned} \tag{4}$$

From Eq. (4), $\beta=\|\mathbf{u}^*\|$ is the so-called Hasofer-Lind reliability index, which comes to be the distance between $\mathbf{u}^*$ and the origin of the standard space. FORM hence consists in finding the design point $\mathbf{u}^*$ and approximating the original limit state function $g_{\mathbf{u}}(\mathbf{u})$ by a tangent hyper-surface at the design point. The first-order approximation of the failure probability becomes:

$$P_f(\mathbf{d}) = P\left[g_{\mathbf{u}}\left(\mathbf{U},\mathbf{d}\right) \le 0\right] \simeq \Phi\left(-\beta\right) \tag{5}$$

Many algorithms can be used for solving the constrained optimization problem of Eq. (4) in order to find the design point. The iHLRF algorithm is used in this paper.

## 2.2 Risk optimization problem

The life-cycle cost of a structural system subject to uncertainties can be decomposed in an initial or construction cost, cost of operation, cost of inspections and maintenance, cost of disposal and expected costs of failure ($C_{expected}$) The expected cost of failure, or failure risk, is given by the product of failure cost ($C_{failure}$) by a failure probability [20]:

$$C_{\text{expected}}(\mathbf{d}) = C_{\text{failure}}(\mathbf{d}) P_f(\mathbf{d}) \tag{6}$$

Failure costs include the costs of repairing or replacing damaged structural members, removing a collapsed structure, rebuilding it, cost of unavailability (downtime), cost of compensation for injury or death of employees or general users, penalties for environmental damage, etc. All failure consequences have to be expressed in terms of monetary units, which in some cases involves considering past compensation.

For each structural component or system failure mode, there is a corresponding failure cost term. The total (life-cycle) expected cost of a structural system becomes:

$$\begin{aligned} C_{\text{total expected}}(\mathbf{d}) = {} & C_{\text{manufacturing}}(\mathbf{d}) + C_{\text{operation}}(\mathbf{d}) \\ & + C_{\text{inspec. \& maint.}}(\mathbf{d}) + C_{\text{disposal}}(\mathbf{d}) \\ & + \sum_{\text{failure modes}} C_{\text{failure}}(\mathbf{d}) P_f(\mathbf{d}) \end{aligned} \tag{7}$$

Any change in **d** that affects cost terms is likely to affect the expected cost of failure. Changes in **d** which reduce costs may result in increased failure probabilities, hence increased expected costs of failure. Reduction in expected failure costs can be achieved by targeted changes in **d**, which generally increase costs. This compromise between safety and cost is typical of structural systems.

A proper point of compromise between safety and cost can be found by solving the so-called risk optimization problem:

$$\mathbf{d}^* = \arg\min[C_{\text{total expected}}(\mathbf{d}) : \mathbf{d} \in \mathbf{S}] \tag{8}$$

where $\mathbf{S}=\{\mathbf{d}_{\min} \leq \mathbf{d} \leq \mathbf{d}_{\max}\}$ is a set of restrictions on the design variables.

Risk optimization can be achieved by controlling failure probabilities and/or the costs of failure. Although failure costs are constant, it is important to note that the present formulation does allow for a trade-off between "competing" failure modes with distinct failure costs. Hence, service and ultimate limit states, with their intrinsically different costs of failure, are readily accounted for in the risk optimization formulation.

## 3 ARTIFICIAL NEURAL NETWORK SURROGATE MODELS

### 3.1 Surrogate models

Surrogate models can be used to aid the solution of computationally expensive problems. Surrogate models are applied in many problems of engineering and science. Surrogate models do not (necessarily) bear any physical (or mechanical) relation to the actual problem, but they represent structural response in a simplified way, by means of some functional relationship which is less expensive to compute.

Limited information on the actual structural response is obtained by computing the solution of the most accurate (numerical) model, at carefully selected points. From these computations, surrogate models are constructed, and used to conduct problem solution further.

Different types of meta-models can be employed in solution of structural engineering problems. The performance of meta-models depends on the nature of the problem being solved, as well as on the choice of points were the "exact" solution is computed.

In this article, artificial neural networks meta-models are employed in the solution of a real problem of structural optimization under uncertainty.

### 3.2 Artificial Neural Networks (ANN)

Artificial neural networks are computational models build based on a simplified analogy to the behavior of the human brain. Information is processed by small processing units (the neurons) which communicate with each other by means of weighted connections (synapses).

Different networks can be constructed by choos-ing different numbers of neuron layers and the type and number of neurons in each layer. Given a set of input-output data, the network is trained: the weight of the various neuron connections is adjusted in order for the network to reproduce the input-output data as accurately as possible. Each iteration in this training is called an epoch. Hence, by means of this training, the connection weights are stored and the network becomes able to predict the response for other input points not present in the training set.

It is common procedure to separate input-output data in two groups: training and validation data. The validation process consists in verifying network error for the validation points, for which the exact response is known. Validation is used between epochs, in order to halt the training process if the network error starts to increase due to overfitting. Separation of the input-output data in training and validation sets reduces the amount of information available for training the network, but helps to avoid problems of overfitting. In the present implementation, 10% of the available input-output information was employed in the validation set (the validation data is always randomly selected).

Some input-output information can also be separated to be used as test set, in order to verify the networks predicting capacity at the end of training. However, in the present application the network is updated in an interactive manner; hence no testing set was used.

Two types of ANN were implemented in this article: Multi-layer Perceptron or MLP and Radial Basis Function or RBF. MLP is generally better to represent the global response, whereas RBF generally yields better local approximations. Details of these two neural networks are presented in the sequence.

### 3.3 Multi-Layer Perceptron ANN

The Multi-Layer Perceptron (MLP) is a type of feed-forward artificial neural network, built with one entry layer (one neuron for each input parameter), one output layer (one neuron for each output parameter) and an arbitrary number of intermediate (hidden) layers.

Multilayer feed-forward networks with as few as a single hidden layer can approximate any arbitrary function and its derivatives through an appropriate choice of activation functions and number of neurons in each layer. Hence, in the present implementation only one hidden layer was employed. The number of neurons ($n_2$) in this layer follows the Kolmogorov-Nielsen theorem:

$$n_2 = 2\left(n + m\right) + 1 \tag{9}$$

where m is the number of optimization variables and n is the number of random variables of the problem. Hence, the number of neurons in the hidden layer is equal to twice the number of input parameters plus one. As the only output parameter is the value of the objective function, the output layer contains a single neuron.

The type of neuron in each layer is defined by the chosen activation functions. Linear activation functions were used for the input and output layers, and tangent-sigmoid neurons were chosen for the hidden layer. In feed-forward networks, data is transmitted from the input layer to the output layer. Input data is first processed by the neurons in the input layer. Each neuron receives one element of the input vector and transmits an impulse to all neurons of the next layer. As the activation function of the input layer is linear, input values are simply passed on to the neurons in

the next layer. In the passage, each value is multiplied by a given weight, which characterizes the connection between neurons of the layers; hence a weighted sum is passed to the hidden layer neurons. A bias factor is added to the weighted sum which allows, for instance, a neuron to be activated even when a null value is passed to it. The resulting value is processed by the activation function of the neuron, returning a value between -1 and +1 (case of the tangent sigmoid function), which is transmitted to the output neuron. The output neuron adds its own bias to the weighted sum he receives and returns the output value for the input data provided to the network. The weights which characterize the connection between the neurons and the bias of each neuron are the network parameters to be determined when the network is trained.

Let $w_{ij}^{(k)}$ be the weight between neuron i of layer k and neuron j of layer k-1, $\theta_i^{(k)}$ be the bias factor of neuron i of layer k, and $\mathbf{y}^{exp}$ be a vector of input data. Then, the output of a three-layer MLP network with $n_k$ neurons in the $k^{th}$ layer is, for this case:

$$y^{net} = \theta_1^{(3)} + \sum_{i=1}^{n_2}\left( w_{i1}^{(3)} \cdot \text{tansig}\left( \theta_i^{(2)} + \sum_{j=1}^{n_1} w_{ji}^{(2)} \cdot y_j^{exp} \right)\right) \tag{10}$$

where $y^{net}$ is the network output and tansig(.) is the tangent sigmoid function.

In this paper the MLP network is trained by using the so-called Levenberg-Marquardt method, a modification of the Gauss-Newton method based on the least squares method for nonlinear models. Like quasi-Newton methods, this method was designed to approach second-order training speed using an approximation of the Hessian matrix, without calculation of the Hessian matrix itself. The parameters of the ANN are adjusted in the opposite direction to the gradient of the error function, in an attempt to decrease this error at each iteration (epoch).

### 3.4 Radial Basis Function ANN

Radial Basis Function networks (RBF) are always formed by three neuron layers: input, intermediary and output layers. RBF are also feed-forward networks: information is transmitted from the neurons in one layer to all neurons in the next layer. In RBF, each neuron in the intermediate layer has a radial basis function as its activation function, whereas neurons on other layers have linear activation functions. The Gaussian activation function is frequently adopted in the literature and is also adopted in the present study.

Let $w_{ij}^{(3)}$ be the weight between neuron 1 of layer 3 and neuron i of layer 2, $cc_i$ and $\sigma_i$ be the central value and the standard deviation of the $i^{th}$ neuron of the intermediate layer, and $\mathbf{y}^{exp}$ be a vector of input data. Then, the output of an RBF network with $n_k$ neurons in the $k^{th}$ layer is:

$$y^{net} = \theta_1^{(3)} + \sum_{i=1}^{n_2} w_{i1}^{(3)} \cdot \exp\left( -\frac{1}{\sigma_i^2} \sum_{j=1}^{n_1} \left( y_j^{exp} - cc_i \right)^2 \right) \tag{11}$$

where $cc_i$ and $\sigma_i$, as well as weights $w_{i1}^{(3)}$ and bias $\theta_1^{(3)}$ are the network parameters to be determined during training.

The number of neurons in the input and output layers are determined by the dimensions of the input and output vectors, respectively. The number of neurons in the intermediate layer is flexible, and it was chosen in such a way as to result in nearly the same number of network parameters of the MLP network. Hence, the computational cost for each training epoch is equivalent for both networks. The number of training epochs, however, is controlled by the validation test, and can be different for each network. In choosing the number of neurons in the intermediate layer, the size of the training data set was also considered, in order to avoid overfitting. Hence, the number of

neurons in the intermediate layer of the RBF network was set as the smallest value between 3/4 of total dataset size and 5/6 of the size of the training data set. The main objective was to avoid having more neurons than data points.

Training of a RBF network is made in two stages: first, the centers and standard deviations of the radial basis functions are found, in such a way that each neuron is trained to represent a given region of the design space; then, the weights and bias are found, calibrating the network output and reducing its prediction error.

In the present article, several training methods were attempted: system of linear equations, generalized delta rule, among others. Good results were obtained with training of the intermediate layer using the k-means clustering method; weights and bias were adjusted using the Levenberg-Marquardt method.

## 4 HYBRID PARTICLE SWARM ALGORITHM

In this article, a hybrid BFGS - Particle Swarm Optimization (PSO) algorithm is used to find the global optimum. The hybrid algorithm is constructed in such a way as to combine the robustness of PSO in finding the global minimum, with the efficiency of the BFGS algorithm in finding the local minimum, once the region of the global minimum is identified. Hence, the search starts with the PSO algorithm, and the region where the global minimum is located is identified. A transition strategy is used to stop the heuristic algorithm and proceed with the local optimization. The detailed algorithm is described in Beck & Gomes [20]. However, in that article the Powel method is used as a local optimizer; in the current implementation, the BFGS method is used as local optimization algorithm.

There are different ways in which a surrogate model can be employed in order to aid solution of an optimization problem. For instance, the surrogate model can be used to replace the mechanical response of the structure or it can be used to replace the objective function. Many authors choose the first option, but in this work, surrogate models are used directly to represent the objective function (Eq. 7). When using surrogate models to approximate the mechanical response, the approximation error can be propagated in evaluation of the objective function; as a result, one has no control of the error in the objective function. Using surrogate models to approximate the objective function directly, one has a proper estimate of the error in the objective function. Moreover, the computational cost of training the networks is justified by the fact that each call to the surrogate objective function avoids several calls to the mechanical model. Recall that the parameters of a surrogate model for the objective function, in risk optimization problems, are the optimization variables $\mathbf{d}$ and the random variables $\mathbf{X}$.

The size of the ANN training data set is an important variable, which was found to significantly affect accuracy and efficiency of the response. The size of the data set must be pre-defined by the user, and basically depends on the complexity of the model being represented (more complex models requiring larger data sets). The size of the training data set is independent of the PSO population size.

Some authors choose to use the surrogate model only after the first heuristic/global optimization iteration is complete. In this paper, once the first 100 data points are obtained, the network is trained and the error is evaluated. If the estimated error is below the acceptable value (0.05), the network is ready to be used by the algorithm: objective function evaluations are replaced by evaluations of the neural network surrogates. Hence, the network can be employed even in the first PSO iteration.

For updating the position of each particle, objective functions can be evaluated in three ways: using one of the two surrogate model approximations or the "exact" objective function. If the ANN error is below the acceptable value, than the objective function approximation at the new position is

the solution provided by the network presenting the smallest (estimated) validation error. If the new objective function value is smaller than the current value for that particle (particle solution is improving), than the surrogate solution is dismissed and the "exact" objective function is evaluated. If not, than the surrogate solution is accepted. This increases accuracy of the network surrogate, as the network will be re-trained using exact solutions which are likely closer to local or global optima. This also increases robustness of the optimization algorithm, as it avoids a particle to be influenced by a solution which was evaluated only through the surrogate model. In addition, a small percentage (20% in this paper) of the candidate solutions are randomly selected to be evaluated in exact form.

Training of the neural networks is made iteratively, during the search for the global optimum. For each new set of ten exactly computed data points, the networks are re-trained, increasing the size of the training set. Once the user-specified (maximum) training set size is achieved, the ten worst data points from the training set are discarded, and replaced by the ten new points. However, new data points are only included in the training set if their objective function values are smaller than the objective function values of the existing training set. This warrants that the surrogate models will have better accuracy at the regions of interest (close to local or global minima). For the initial training, one hundred training epochs are allowed; for the remaining trainings only five epochs are allowed. However, training can be interrupted earlier, by force of the validation process.

Since the output range of the tangent-sigmoid function is [-1 1], output data has to be mapped to match output range of the physical problem. In this paper, the normalizing mapping procedure for minima and maxima was used. Validation and training errors were evaluated in transformed space, as the root-mean square errors for the respective samples. The acceptable validation error was set to 0.05.

Neural networks are used in the following to aid solution of global optimization problems. Some local optimization problems are also solved, using the BFGS algorithm; for these problems no networks are used.

## 5 EXAMPLE: POWERLINE TOWER SUBJECT TO RANDOM WIND LOADS

### 5.1 Problem definition and limit state

The problem considered herein is based on a problem solved by Hansen & Vanderplaats [29]. A steel frame transmission line tower is optimized with respect to its configuration (nodal positions) and member size (cross-sectional area). Hansen & Vanderplaats [29] considered only the loads originating from the weight of the cables. However, one of the largest sources of uncertainty in the behavior of such towers arises from random wind loads. Hence, random wind loads are incorporated in the current analysis.

In ref. [29] the mechanical problem was modeled using truss elements, with restrictions in terms of tensile stresses or Euler buckling stresses. In the present paper, beam elements with L-shaped cross sections are used. A positional finite element formulation is employed, which allows physical and geometrical non-linearities to be considered. The beam elements used have three nodes: two extreme and a central node and 3 d.o.f. per node. The material is assumed elastic-perfectly plastic, and the limit state function for tower collapse is obtained from the load-displacement diagram (Figure 1). The limit state function must be robust with respect to the different structural configurations that are tested by the global optimization algorithm, and must take into account the non-linear behavior before failure. The FE algorithm which was used had no special capabilities to find the limit load from the load-displacement diagram. Hence, the "collapse" failure criterion was written in terms of the angle of the load-displacement curve:

$$g(\mathbf{x},\mathbf{d}) = 75^{o} - atan\left(\frac{\Delta\delta}{\Delta L}\right) \tag{12}$$

where $\Delta\delta$ is an increment in mean displacement (between nodes 17 and 22), measured in cm, and $\Delta L$ is an increment in the non-dimensional load factor, which is calculated from mean loads. In Figure 1 it can be observed that this does not quite correspond to the collapse load, but it is a robust criterion which takes into account the non-linear behavior before collapse. For critical angles larger than $75^{o}$, numerical difficulties were encountered.

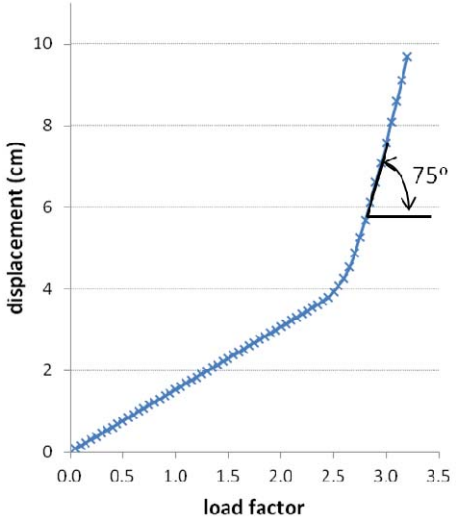


Figure 1: Load-displacement diagram for the tower (load factor x tip displacement).

## 5.2 Desing variables and random variables

The finite element model of the tower contains 47 beam elements and 69 nodes, with 207 total number of degrees of freedom. This particular FE model is not too expensive to evaluate, but it is a proper model to solve the proposed real structural engineering problem. Optimization variables are the *x* and *y* coordinates of the nodes, and the cross-sectional areas of the elements. The nodes holding the cables are fixed; hence their coordinates are not optimization variables. Moreover, tower symmetry is considered in definition of the optimization variables, since the direction of the wind action is arbitrary. Hence, the problem contains 44 optimization variables and m=dim[**d**]=44.

As mentioned, solving global optimization problems presents an extra difficulty which is the evaluation of (FE) responses for structural configurations which are far from conventional. This concern must be balanced with the necessity of imposing the least restrictions possible, in order to render the obtained solution a true global optimum. Our concern for imposing the least restrictive bounds on the design variables, but still rendering the numerical models tractable, resulted in the following bounds for the optimization variables: 50% less and 25% more for the cross-sectional areas and ±20 cm for the nodal coordinates (with reference to the initial configuration).

Another particularity of applying heuristic methods for the solution of real engineering problems (large number of design variables and high computational cost) is that it is impossible to cover the whole design space, due to processing time limits, convergence problems and numerical instabilities. Hence, initialization methods different to the usual are required. In this study, PSO particles were initialized by applying random perturbations within the bounds for each of the design variables. This way, different structural configurations are tested, but none of them too far from conventional design configurations.

### 5.3 Objective functions and preliminary results

The problem was solved in several steps, in order to provide the best possible initial configuration for the more demanding optimizations tasks. Hence, the problem was first solved with the objective function of minimizing volume of steel ($C_{ini}$), starting with the optimal configuration of ref. [29]:

$$C_{ini}(\mathbf{d}) = \sum_{i=1}^{n_{elem}} V_i(\mathbf{d})\rho\$; \quad C_{ref} = C_{ini}(\mathbf{d}_{\text{Vanderplaats}}) \tag{13}$$

where $V_i$ is volume of the $i^{\text{th}}$ element, ρ is steel density and \$ is the unit cost of material. The initial, reference cost ($C_{ref}$) was assumed as the cost of steel for the Vanderplaats [29] configuration. The cost of one kg of steel was assumed as one monetary unit. This first problem was solved using only the BFGS algorithm (BFGS_DET), and using a safety coefficient of two for the critical displacement.

For the risk optimization problem, the objective function is the minimization of total expected costs. This includes initial costs and expected costs of failure ($C_{ef}$) which is evaluated as:

$$\mathrm{C}_{\mathrm{ef}}(\mathbf{d}) = \left(100\,C_{ref} + 2\,C_{ref} + C_{ini}(\mathbf{d})\right) P_f(\mathbf{d}) \tag{14}$$

where $C_{ini}$ is the initial cost (of re-building the failed tower), the cost of removal of the failed tower is $2C_{ref}$ and the penalty for interruption of energy supply is $100C_{ref}$. The initial cost is a function of the optimization variables (nodal coordinates and cross-sectional areas). The total expected cost is given as:

$$\mathrm{C}_{\mathrm{et}}(\mathbf{d}) = \mathrm{C}_{\mathrm{ef}}(\mathbf{d}) + \mathrm{C}_{\mathrm{ini}}(\mathbf{d}) \tag{15}$$

The second optimization problem is solved only using the BFGS method (BFGS_RISK).

Starting from the solution of the local deterministic optimization problem (BFGS_DET), four global optimization runs where obtained, all using the hybrid (PSSB) algorithm and:

1. no surrogate models (PSSB_noSUR);
2. the RBF surrogate model only (PSSB_RBF);
3. the MLP surrogate model only (PSSB_MLP);
4. both surrogate models (PSSB_SURR).

### 5.4 Overview of results

The solutions to the local deterministic (BFGS_DET), local risk (BFGS_RISK) and global structural optimizations are compared Table 1. In Tables 1, the abbreviations are as follows: value of objective function (OBJ_F), number of calls to mechanical model (NC_ME) and computation time (TIME) in seconds.

The local deterministic optimal configuration has a steel mass of 837 kg, a failure probability of $8.91\ 10^{-4}$, and total expected cost of 914 monetary units (m.u.), as shown in Table 1. This configuration was used as starting point for all the other optimization analyses.

The optimum results for the local risk optimization problem (BFGS_RISK) are also shown in Table 1. In comparison with the local deterministic solution, the risk optimization optimum has wider columns but narrower braces and battens. The total expected cost for the optimum risk optimization solution is 799 m.u., hence 5% less than for the deterministic volume minimization problem.

Table 1. Results for local and global risk optimization analyses, with and without ANN surrogates.

| | OBJ_F | NC_ME | TIME (sec) |
|---|---|---|---|
| BFGS_DET | 837 | 479 | 15 |
| BFGS_RISK | 799 | 21,034 | 625 |
| PSSB_noSUR* | 726( -9%)† | 96,648 | 14,276 |
| PSSB_MLP* | 712(-11%)† | 64,854(-33%)ד | 11,475(-20%)ד |
| PSSB_RBF* | 716(-10%)† | 67,110(-31%)ד | 7,120(-50%)ד |
| PSSB_SURR* | 730( -9%)† | 62,379(-35%)ד | 8,770(-39%)ד |

*Mean values for three runs.
† reduction with respect to BFGS_RISK solution;
ד reduction with respect to PSSB_noSUR solution.

It can be observed in Table 1 that results for the global risk optimization analysis are 9% better than results for the local risk optimization, in terms of objective function value. This result by itself justifies use of global optimization algorithms. However, this improvement comes at a significant increase in computational cost. In the global solution, the number of calls to the mechanical model increased 359% and the computation time increased 2184% with respect to the local risk optimization solution. The computation time increases much more than the number of calls to the mechanical model because each non-linear finite element solution requires several load steps for convergence of the Newton-Raphson algorithm. Moreover, the global PSO search tests weird structural configurations, which take more Newton-Raphson interactions for convergence than more conventional structural configurations. The challenge of using ANN surrogates for the objective function response is to obtain the same reduction in objective function but at a fraction of the computational cost of the direct solution using no surrogates.

The optimum structural configuration for the PSSB_noSUR solution is presented in Figure 2.

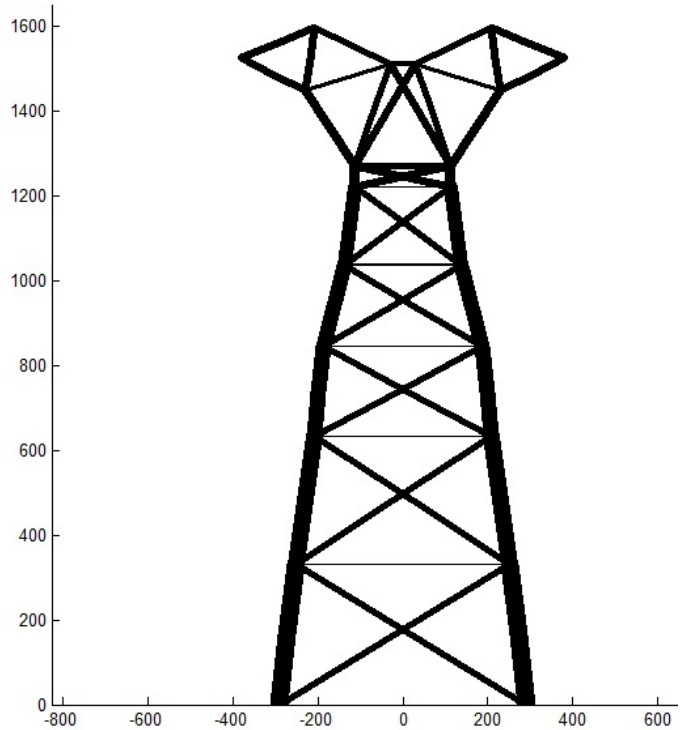


Figure 2: Optimum structural configuration for global risk optimization solution (PSSB_noSUR).

### 5.5 Results using ANN surrogates with 150 training points

Results for the three global risk optimization runs using ANN surrogates were found to be dependent on the size of the training data sets of the surrogate models. Clearly, more training points add to the accuracy of the surrogate model representation, but could also lead to worst performance, since the training process becomes more involved. Hence, the ideal size of the training data set does not depend only on the complexity of the system response being modelled, but also on the computational cost of the *exact* numerical solution. A complete investigation of the response dependency with respect to the size of the training data set will be subject of future research. In this paper, results are presented for ANN with 150 training points. Results presented herein are the mean values for three runs of each algorithm.

Table 1 summarizes results obtained for the global optimization analysis using ANN surrogates with 150 training points. It can be readily observed that the solutions using the ANN surrogates is more efficient than the original global risk optimization solution with no surrogates. The objective function values are around 10% less than results for the local solution, but computational costs are much smaller. In comparison to the global solution with no surrogates, the third column of Table 1 lists the number of mechanical model calls and the percentage of calls avoided by use of surrogate models, w.r.t. the PSSB_soSUR solution. It is observed that the reduction is not drastic, but significant. There is also an associated reduction in the computation time, although this gain does not vary linearly with the reduction in mechanical model calls. This is due to the computational cost of training the different networks, but also to the fact that not all calls to the mechanical model have the same computational cost. The gain in terms of mechanical model calls is considered more significant than the reduction in computation time because the finite element model of the tower is not too expensive, and the FE computation cost is not much smaller than the costs for training and evaluating the neural networks.

Use of a combination of two surrogate models resulted in a slight gain in performance (reduction of mechanical model calls), when compared to the isolated MLP and RBF networks. However, the reduction in objective function value was not as large for the combined network use.

The optimum structural configuration for the PSSB_SUR solution is presented in Figure 3.

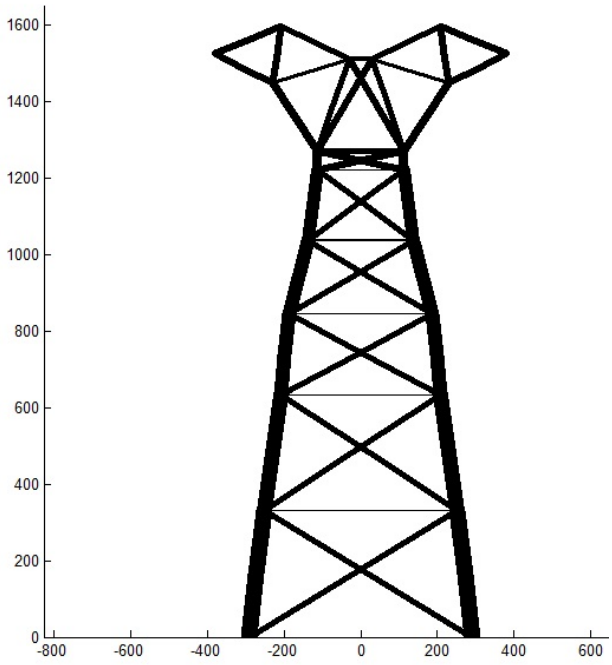


Figure 3: Optimum structural configuration for global risk optimization solution (PSSB_SURR).

## 6 CONCLUDING REMARKS

This article has addressed some challenges in solving real structural optimization problems in consideration of uncertainties. The investigation proposed to advance in four areas often (individually) neglected in the published literature: a. incorporation of geometrical and material nonlinearities in structural response; b. consideration of a proper number of design variables; c. solution for the global minima and d. consideration of the expected consequences of failure. These areas, in combination with large FE models, characterize the scope of real structural engineering optimization problems.

The article presented a practical problem of optimization of a steel frame transmission line tower subject to random wind loads. With respect to the objectives listed above, the total number of d.o.f. was not very large, but still sufficient to illustrate the challenges to solving a real structural optimization problem under uncertainties.

It was shown that, when uncertainties are relevant, optimum (minimum cost) design can only be achieved by considering the expected consequences of failure. This has been called risk optimization. It was argued that risk optimization problems present multiple local minima, hence global optimization algorithms become necessary. Indeed, the risk optimization problem was first solved using a local optimization algorithm. Starting from this local optimum, the objective function was reduced by another 10% by solving the global optimization problem. The computational cost, however, increased dramatically.

Artificial Neural Networks were employed as surrogates for the objective function, significantly reducing the computational effort. Reductions of the order of 23 to 31% were obtained in the required number of mechanical model calls by using RBF networks. MLP networks led to larger reductions, between 33 and 48%. Using a combination of both networks led to reductions between 35 and 43% in the required number of mechanical model calls. These reductions in computational effort caused no prejudice to global optimum objective function values.

It was observed that the performance of ANN surrogates is highly dependent on the size of training data sets. A throughout investigation of these effects will be the subject of future research.

## ACKNOWLEDGEMENTS

Sponsorship of this research project by the São Paulo State Foundation for Research - FAPESP (grant number 2009/17365-6) and by the National Council for Research and Development - CNPq (grant number 301679/2009-6) is greatly acknowledged.

## REFERENCES

[1] Qu X, Hafta RT, 2003: "Design under uncertainty using Monte Carlo simulation and probabilistic factor". Proceedings of ASME DETC'03 Conference, Chicago, IL.

[2] Ching J, Wei-Chih H, 2008: Transforming reliability limit-state constraints into deterministic limit-state constraints", Structural Safety 30 (2008), 11–33.

[3] Du X, Chen W, 2004: "Sequential Optimization and Reliability Assessment method for Efficient Probabilistic Design". ASME J. Mech. Des., 126(2), 225–233.

[4] Tu J, Choi KK and Park YH, 1999: "A New Study on Reliability-Based Design Optimization", J. Mech. Des. 121 (4), 557-565.

[5] Cheng GD, Xu L, Jiang L, 2006: "A Sequential approximate programming strategy for reliability-based optimization". Computers and Structures, 84(21), 1353–1367.

[6] Youn BD, Choi KK, 2004: "A new response surface methodology for reliability-based design optimization", Computers and Structures 82, 241–256.

[7] Aoues Y, Chateauneuf A, 2010: "Benchmark study of numerical methods for reliability-based design optimization". Struct Multidisc Optim, 41:277–294.

[8] Valdebenito MA, Schueller GI, 2010: "A survey on approaches for reliability-based optimization", Struct. Multidisc. Optim 42:645-663.

[9] Schuëller GI and Jensen HA, 2008: "Computational methods in optimization considering uncertainties – An overview", Computer Methods in Applied Mechanics and Engineering, Vol. 198, Issue 1, 2-13.

[10] Beyer HG, Sendhoff B, 2007: "Robust optimization - A comprehensive survey", Computer Methods in Applied Mechanics and Engineering, 196, 3190-3218.

[11] Enevoldsen I, Sorensen JD, 1994: "Reliability-based optimization in structural engineering", Structural Safety 15, 169-196.

[12] Soltani M, Corotis RB, 1988: "Failure cost design of structural systems", Structural Safety 5, 239-252.

[13] Aktas E, Moses F and Ghosn M, 2001: "Cost and safety optimization of structural design specifications", Rel. Eng. and Sys. Safety 73, 205-212.

[14] Frangopol DM and Maute K, 2003: "Life-cycle reliability-based optimization of civil and aerospace structures", Computers & Structures 81, 397-410.

[15] Streicher H, Rackwitz R, 2004: "Time-variant reliability-oriented structural optimization and a renewal model for life-cycle costing", Probabilistic Engineering Mechanics 19, 171-183.

[16] Bucher C, Dan M. Frangopol DM, 2006: "Optimization of lifetime maintenance strategies for deteriorating structures considering probabilities of violating safety, condition, and cost thresholds", Probabilistic Engineering Mechanics 21, 1-8.

[17] Haukaas T, 2008: "Unified reliability and design optimization for earthquake engineering", Probabilistic Engineering Mechanics 23, 471-481.

[18] Mínguez R, Castillo E, 2009: "Reliability-based optimization in engineering using decomposition techniques and FORMS", Structural Safety 31, 214-223.

[19] Beck AT, Verzenhassi CC, 2008: "Risk Optimization of a Steel Frame Communications Tower Subject to Tornado Winds", Latin Am. Journal of Solids and Structures 5, 187-203.

[20] Beck AT, Gomes WJS, 2012: "A comparison of deterministic, reliability-based and risk-based structural optimization under uncertainty", Probabilistic Engineering Mechanics 28, 18-29, 10.1016/j.probengmech.2011.08.007.

[21] Beck AT, Gomes WJS, Bazan FAV, 2012: "On the Robustness of Structural Risk Optimization with Respect to Epistemic Uncertainties". Int. J. for Uncertainty Quantification 2, 1-20, 10.1615/Int.J.UncertaintyQuantification.v2.i1

[22] Gomes WJS, Beck AT, 2011: "Hybrid Algorithms Based on Particle Swarm Optimization and the Powell Method for Global Optimization". Proceedings of the Thirteenth International Conference on Civil, Structural and Environmental Engineering Computing. Stirlingshire, Scotland : Civil-Comp Press.

[23] Coda HB, Greco M, 2004: "A simple FEM formulation for large deflection 2D frame analysis based on position description". Computer Methods in Applied Mechanics and Engineering, 193, 3541-3557.

[24] Greco M, Coda HB, 2006: "Positional FEM Formulation for flexible multi-body dynamics analysis". Journal of Sound and Vibration 290, 1141-1174.

[25] Greco, M., Gesualdo FAR, Venturini WS, Coda HB, 2006: "Nonlinear positional formulation for space truss analysis". Finite Elements in Analysis and Desing 42, 1079-1086.

[26] Coda HB, Paccola RR, 2007: "An alternative positional FEM formulation for geometrically non-linear analysis of shells: curved triangular isoparametric elements". Comput. Mech. 40, 185-200.

[27] Coda HB, Paccola RR, 2010: "Improved finite element for 3D laminate frame analysis including warping for any cross-section", Applied Mathematical Modelling 34, 1107–1137.

[28] Coda HB, Paccola RR, 2011: "A FEM procedure based on positions and unconstrained vectors applied to non-linear dynamic of 3D frames", Finite Elements in Analysis and Design 47, 319–333.

[29] Hansen SR & Vanderplaats GN, 1990: "An approximation method for configuration optimization of trusses", The AIAA Journal 28, 161-172.