Paper 203

Proceedings of the Eleventh International Conference
on Computational Structures Technology,
B.H.V. Topping, (Editor),
Civil-Comp Press, Stirlingshire, Scotland

An Open Computational Framework for Reliability Based Optimization

E. Patelli and M. de Angelis
Centre for Engineering Sustainability, School of Engineering
University of Liverpool, United Kingdom

Abstract

This paper presents an open computational framework for reliability based. The framework has been designed to provide the maximum flexibility allowing the state of the art in reliability analysis (e.g. adopting advanced Monte Carlo methods) to be combined in the direct approach as well as in the construction of the different type of meta-models (e.g. response surface, artificial neural networks, kriging model and polynomial chaos, etc.). A set of widely used gradient-based and gradient-free optimization algorithms are also available for performing the optimization step as well as high performance computing capability. Numerical applications show the applicability and flexibility of the proposed framework for solving real-life problems.

Keywords: reliability based optimization, matlab, open source, high performing computing, meta modelling, numerical methods.

1 Introduction

In nowadays engineering practice, optimization is almost an indispensable step of the design cycle for any product/component. Optimize means design a better product or system that can reach significant reductions in terms of the manufacturing and operating costs, as well as the improvement in the performance. However, these products are affected by uncertainties, caused by lack of sufficient knowledge and/or by natural unpredictable external events. Ignoring the effects of the uncertainties the "optimized" products can perform unsatisfactory in realistic conditions, for instance they can show a very low reliability, high reparation and maintenance costs etc.

In order to cope with this problem and to guarantee that the components or systems will continue to perform satisfactory despite fluctuations and changes of model (e.g. due to production processes) and environmental conditions (e.g. due to climate change), the design has to be "robust". Consequently, the field of optimization has been coupled with reliability analysis (see e.g. [18]) forming the so-called reliability based optimization analysis (see e.g. [12, 21, 5]).

Here, an open computational framework for reliability based optimization is presented. Developed in a object oriented fashion in Matlab environment, this framework provides the necessary flexibility, modularity and usability to be adopted in different contexts. Thanks to the terms of the LGPL license [1] adopted, anybody are allowed to use, verify and modify the proposed framework or derived code from it. This framework is used by the general purpose software COSSAN-X [16].

This framework allows to perform reliability based optimization adopting the direct approach, global and local surrogate models. It allows to combine the state of the art in reliability analysis (e.g. adopting importance sampling, line sampling [13]) in the direct approach as well in the construction of the meta-models (e.g. response surface, artificial neural networks, kriging model and polynomial chaos [7]). A set of widely used gradient-based and gradient-free optimization algorithms (e.g. SQP, Cobyla, genetic algorithms, simulated annealing etc. [3]) are also available. Furthermore, adopting the high performance computing capability (grid and cloud computing), the proposed approach allows the analysis of realistic problems. The applicability and the flexibility of the proposed framework for solving real-life problems will be demonstrated by means two applications considering static and dynamic load.

The outline of the present paper is as follows: Section 2 reports a brief overview of the computationally challenging in the reliability based optimization problems. In Section 3, the computational framework for solving efficiently a large range of reliability based optimization problems is described. In Section 4, application examples are presented to demonstrate the applicability of the computational framework for solving problems of practical interest. Finally, some final remarks are listed in Section 5.

2 Reliability Based Optimization

The reliability based optimization approach is an attractive and most useful design tool: it allows to determine the best design according to some predefined criteria. The main aim of the reliability based optimization is to consider explicitly the effect of the uncertainties in the optimization problem. In fact, in any practical situation there are a number of model parameters which are not known at the design stage and that might affect the performance of the products or systems. These parameters that might be affected by uncertainty can be characterized as random variables θ. The rational quantification of the effects of these uncertainties on the product/system performance requires an appropriate model to measure the plausibility of a given realization of θ. The performance space of the component/system under investigation can be split in two parts by means of the so-called performance function g. The performance function define a safe (admissible) performance and a failure (inadmissible) performance based on certain performance requirements, e.g. the demand exceeds the capacity of the system. It should be noted that, in this context, failure does not necessarily imply collapse but rather an undesirable performance.

The performance depends generally on the values of the design variables ($\boldsymbol{x}$) and the uncertain parameters ($\boldsymbol{\theta}$), i.e. $g = g(\boldsymbol{x}, \boldsymbol{\theta})$. The performance function is defined such that $g(\boldsymbol{x}^{+}, \boldsymbol{\theta}^{*})$ is smaller or equal to zero when a specific realization of $\boldsymbol{\theta}^{*}$, in combination with a specific set of design variables $\boldsymbol{x}^{+}$, causes failure (i.e. an unacceptable performance of the system); in case of an acceptable performance, $g(\boldsymbol{x}^{+}, \boldsymbol{\theta}^{*})$ is larger than zero. It is important to note that a realistic reliability model of a system may involve the definition of several failure events. The probability of failure computed solving the following integral:

$$p_F(\boldsymbol{x}) = \int_{g(\boldsymbol{x},\boldsymbol{\theta})\leq 0} f(\boldsymbol{\theta})\, d\boldsymbol{\theta} \tag{1}$$

where $f(\boldsymbol{\theta})$ represents the joint probability density function of the uncertain factors.

In practical situations, several different design solutions can satisfy prescribed performance objectives. The design variables are those parameters that can be adjusted and tune in order to improve the performance of a component or system. Typical examples of design variables are the cross sections of structural members, interval of inspection and repair, topology parameters, set of components, etc. The design variables can also represent a characterized (moment) of an uncertain parameter, some examples are the mean time to failure of a component, the admissible tolerance for the specific components (i.e. the variance), etc. Hence, the final solution must be chosen using appropriate criteria. The spectrum of possible goals is rather wide and it is problem dependent, such as minimization of costs, i.e. the chosen solution minimizes construction, maintenance and eventual collapse costs during the life time of the facility. The objective function, denoted by $F(\cdot)$, depends on the design variables and, eventually, on the uncertain parameters, i.e. $F(\boldsymbol{x})$ and $F(\boldsymbol{x}, \boldsymbol{\theta})$, respectively. The optimization step requires the repeated evaluation of the objective function, $F(\cdot)$, and constraints, $C(\cdot)$, for different values of the design variables, $\boldsymbol{x}$, in order to identify the optimal design. In mathematical terms, an optimization problem is formulated as follows:

$$\min\ F(\boldsymbol{x}),\ \ \boldsymbol{x}^T = \langle x_1, x_2, \ldots, x_n \rangle \tag{2}$$

subject to:

$$\boldsymbol{x} \in \Omega_x\!: \quad C_j(\boldsymbol{x}) = 0,\ j = 1, \ldots, m_e \text{ and } C_j(\boldsymbol{x}) \leq 0,\ j = m_e + 1, \ldots, m \tag{3}$$

In an unconstrained optimization problem, all the design space Ω is feasible. One example of unconstrained problem is the optimization of a component/system in order to achieve a prescribed reliability level. In this case the objective function can be defined as the square difference between the prescribed reliability level and the actual reliability level of the component/system, e.g.:

$$F(\boldsymbol{x}, \boldsymbol{\theta}) = (p_F^* - p_F\hat{(\boldsymbol{x})})^2 \tag{4}$$

where $\hat{p_f}$ represents the estimated failure probability of the component and the p_F^* prescribed failure probability.

More often, not all the possible solutions are feasible (e.g., the component might fulfil certain specific design requirements or the total costs are to be below a prescribed threshold). The admissible regions are defined by the so-called constraint functions ($C(\cdot)$). The constraints can be of deterministic nature when they refer exclusively to the design variables $\boldsymbol{x}$. Deterministic constraints are formulated as a mathematical function, $C(\boldsymbol{x})$; usually, a deterministic constraint is defined such that $C(\boldsymbol{x}) \leq 0$ implies the satisfaction of a constraint. The constraints may also include both design variables and uncertain parameters. In such cases, the constraint is probabilistic and its fulfilment will be associated with one (or more) of the failure events defined above $C(\boldsymbol{x}, \boldsymbol{\theta})$. For instance, a probabilistic constraint is satisfied when the probability of occurrence of the failure event ($p_F(\mathbf{x})$) is equal or smaller than a prescribed probability level (p_F^*), i.e. the constraint is satisfied if $p_F(\boldsymbol{x}) - p_F^* \leq 0$ as shown in Section 4.1.

Direct approaches The reliability based optimization analysis can be carried out adopting a so-called direct approach. This means that at each iteration of the optimization loop a full reliability analysis is performed (i.e. estimating the integral of Eq. 1). Nevertheless, different strategies and efficient methods can be adopted to perform efficiently the reliability analysis. For instance, the reliability analysis can be based on approximate method (e.g. FORM [9]), plain Monte Carlo method or performed adopting advanced and efficient Monte Carlo methods such as Line Sampling [20] or Subset [4] simulation.

It is important to notice that the computational cost of the reliability analysis plays a fundamental role in the feasibility and applicability of the direct approach. Moreover, the estimation of the failure probability is usually quite noisy, therefore robust optimization methods such as Cobyla, simplex [3] are needed.

Surrogate models A possible means for reducing the numerical costs associated with the solution of reliability based problems is to replace the computationally expensive part of the reliability based optimization analysis by a surrogate-model. Surrogate-models mimic the behaviour of the original model, by means of an analytical expression with negligible computational cost making the efficiency of the optimization methods almost irrelevant. The approximation is constructed by selecting some predefined interpolation points in the design space, at which the failure probability is estimated; then, a surrogate model is adjusted to the data collected in a least square sense. As the construction of this approximation over the entire domain can be demanding, it may be easier to generate an approximation of the failure probabilities over a sub-domain (see e.g. [14]), i.e. to generate a local surrogate model. Local surrogate model might require generally less evaluation points to be constructed although they have to be continuously updated in order to following the current values of the design variables. The surrogate model can be introduced at two different levels: to replace the model (i.e. the estimation of the performance function) or directly to replace the reliability analysis (i.e. the failure probability).

The most used surrogate model in the reliability based optimization analysis is the response surface methodology introduced by [6] and used a number of times in the literature of reliability based optimization (see e.g. [19]). The advantage of this approach is that the reliability assessment step is decoupled from the optimization problem, i.e. the response surface is inexpensive to evaluate and thus, any appropriate algorithm can be used to solve the optimization problem. Other surrogate models include Artificial Neural Networks, Polynomial Chaos, Polyharmonic Splines [11].

As the construction of this approximation over the entire domain of the design variables can be demanding, it may be easier to generate an approximation of the failure probabilities over a local domain, i.e. to generate a local surrogate-model, see e.g. [23].

3 OpenCossan

OpenCossan is a collections of methods and tools under continuous development carried on at the Institute for Risk and Uncertainty at the University of Liverpool (UK) based on the originally development by the group of Prof. Schuëller at the University of Innsbruck (Austria) [16]. The OpenCossan is open source software released under the LGPL licence [1]; this means that the program can be used for free, redistribute and/or modify under the terms of the GNU General Public License.

These components are developed on Matlab® environment, known for its highly optimized matrix and vector calculations and high level programming environment. These components include several predefined solution sequences to solve a number of different problems. Typical applications include UQ and management, reliability based optimization and robust optimization, life-cycle management, model validation and verification.

It is important to mention that the proposed framework gives the maximum of the flexibility to the users. Thanks to the modular nature of OpenCossan, coded exploiting the object-oriented Matlab programming environment, it is possible to define specialized solution sequences including any reliability method, optimization strategy and surrogate model or parallel computing strategy to reduce the overall cost of the computation without loss of accuracy. Additionally, new reliability methods or optimization algorithm can also be easily added.

The computational framework is organized in classes, i.e. data structures consisting of data fields and methods together with their interactions and interfaces. Objects, that are instances of classes, can be aggregated forming more complex objects and proving methods (i.e. solutions) for practical problem in a compact, organized and manageable format. For instance, to perform the reliability analysis is necessary to define an object of type "Probabilistic Model" that defined the problem under investigation by combining Parameters, Random Variables, Performance Function and a Physical Model (e.g. a FE model) objects. Then, the reliability analysis (i.e. the estimation

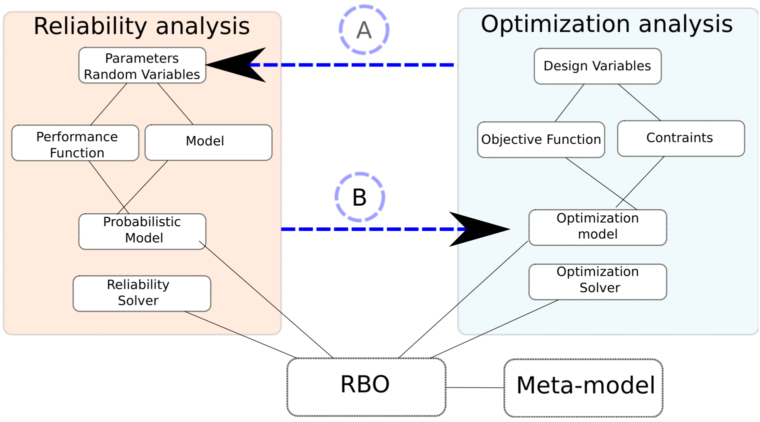

Figure 1: Scheme of the open framework for reliability based optimization. The arrows A and B indicate the dependencies between the reliability toolbox and the optimization toolbox.

of the failure probability) can be performed combining this object with a reliability solver object that defines the reliability methods (e.g. Monte Carlo, Line Sampling etc.): Probabilistic Model Object + Reliability solver object → Failure probability.

The reliability solver allows to estimate the failure probability by means of approximate methods (see e.g. [10]) or advanced simulation-based methods (see e.g. [20]). First and second order reliability method and estimation of bounds can be mentioned within the first category. The family of the advanced simulation-based methods includes importance sampling [17], line sampling [20] and subset simulation [4].

In the case of an optimization analysis, it is necessary to define an object of class "Optimization Model" (that contains other objects used to define the Design Variables, Objective Function and Constraints) plus a solver, i.e. an instance of a class "Optimizer" such as Cobyla, Simplex, Genetic Algorithms, etc. Then the optimization is performed combining these objects: Optimization Model Object + Optimization solver object → Optimum solution. The optimization toolbox provides a set of widely used algorithms to solve constrained and unconstrained, continuous and discrete, standard and large-scale optimization problems. The optimization toolbox includes gradient-based, gradient-free, deterministic and stochastic optimization methods (see Table 1).

Performing a reliability based optimization analysis requires 4 objects: Optimization Model, Optimization solver, Probabilistic Model, Reliability solver as shown in Figure 1. In addition, it is necessary to provide the mapping between the design variables (defined in the Optimization model) and the parameters of the model defined in the Probabilistic Model (see arrow A in Figure 1). An example of a script used in OpenCossan for performing reliability based optimization analysis is shown in Figure 2.

In the reliability based optimization, the computational efforts might become infeasible. One way to reduce the analysis time is to use meta-models, which approximate

```
% ==================================================================
% This file is part of openCOSSAN.  The open general purpose matlab
% toolbox for numerical analysis, risk and uncertainty quantification.
% Author: Edoardo Patelli
% http://cossan.co.uk
% ==================================================================

% Definition of the Model
MyModel= Model('Xevaluator',ModelSteelRoofTruss,'Xinput',MyInput);

% Definition of the Performance Function
MyPerformanceFunction = PerformanceFunction('Sdemand','maxDisp', ...
                    'Scapacity','displacementCapacity','Soutputname','Vg');

% Definition of the Probabilistic Model
MyProbabilistiModel=ProbabilisticModel('Xmodel',MyModel, ...
                             'XperformanceFunction',MyPerformanceFunction);

                         % Definition of the Reliability solver
MySimulator=LineSampling('NmaxLines',20);

% Definition of the Optimization problem
MyDesignVariables =  ...
   Input('CXmembers',{DesignVariableA1,DesignVariableA2,DesignVariableA3});

% Definintion of the objective function
MyObjectiveFunction = ObjectiveFunction('Sscript', ...
     'for n=1:length(Tinput), Toutput(n).fobj=Tinput(n).totVolumeDV; end');

% Definintion of the contraint
MyConstraint = Constraint('Sscript', ...
       'for n=1:length(Tinput), Toutput(n).fcon=(Tinput(n).pf-1e-4); end');

% Definition of the RBO problem and the mapping between Design Variable and
% Model parameters
MyRBOProblem=RBOProblem('XprobabilisticModel',MyProbabilistiModel, ...
                        'Xsimulator',MySimulator', ...
                        'Xinput',MyDesignVariable, ...   % Design Variables
                        'XobjectiveFunction',MyObjectiveFunction,...
                        'Xconstraint',MyConstraint,...
                        'SfailureProbabilityName','pf',... % Name of the pf
                     ... % Mapping
            'CdesignvariableMapping',{'ObjectA1' 'A1' 'parametervalue';...
                                      'ObjectA2' 'A2' 'parametervalue';...
                                      'ObjectA3' 'A3' 'parametervalue'});
% Definition of the Optimization solver
MyOptimizer=COBYLA('NmaxIterations',50);

%% Performing RBO analysis
Xoptimum=MyRBOProblem.optimize('Xoptimizer',MyOptimizer);
```

Figure 2: Example of a script used for the reliability based optimization.

the quantity of interest at low computational cost. Meta-models mimic the behaviour of the original model, by creating input-output relations that approximate the real one by means of basic mathematical operations. Different surrogate models exist to approximate generic input-output relations, such as response surfaces [15] and Artificial Neural Networks (ANN) [2]. An important indicator of the goodness of a meta-model after the training (i.e. the calibration of the meta-model) is the coefficient of determination R^2, defined as [8]:

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \overline{y}_i)^2} \tag{5}$$

where y_i are the outputs of the full model, $\overline{y}_i = \frac{1}{N_{data}} \sum_{i=1}^{N_{data}} y_i$ and $\hat{y}_i$ are the outputs predicted by the meta-model. The accuracy of the output prediction of the meta-model can be judged by the closeness of the value R^2 to the target value of 1.0, which expresses an exact match of the surrogate model prediction and the output of the full model.

Adding an object of type "meta-model" to the "RBO object", the analysis is performed adopting the meta-model instead the reliability analysis block to estimate the failure probability. In order to address these problems efficiently, the software includes the state-of-the-art of the algorithms and numerical procedures for simulation, reliability and optimization analysis, respectively. Table 1 shows the main algorithms and procedures implemented in OpenCossan.

Table 1: Main classes available for the reliability based optimization available in the OpenCossan. In italic the classes that have shown to be more efficient for solving reliability based optimization problems.

Toolbox	Main algorithms and procedures
Reliability	Monte Carlo, LatinHyperCube Sampling, Sobol' Sampling, Halton Sampling, *Line Sampling*, Subset simulation and approximate methods (*FORM*, bounds)
Optimization	*BFGS*, *COBYLA*, Cross Entropy, Evolution Strategies, Genetic Algorithms, MiniMax, *Simplex*, Simulated Annealing, *SQP*, Stochastic Ranking
Meta-modelling	*Artificial Neural Networks*, *Response Surface*, *Polyharmonic Splines*, polynomial-chaos

4 Application examples

4.1 Steel Roof Truss

Description of the problem In this numerical example, the linear static behaviour of a steel roof truss is herein analysed. The aim is to optimize the total volume of the structure, i.e. the quantity of material required for constructing the steel roof truss

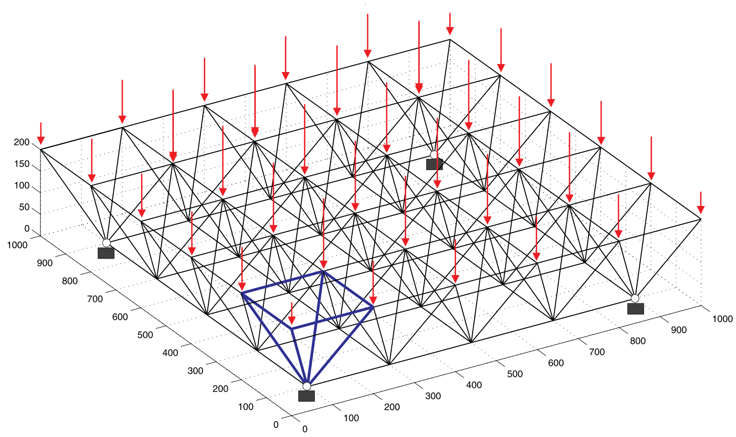

Figure 3: Scheme of the steel roof truss and the load applied.

taking into account the effect of the uncertainties. It is imposed as a constraint of the optimization problem that the failure probability has to be lower than 10^{-4}. System failure is defined as the exceedance of the maximum allowed nodal displacement defining the performance function.

The steel roof truss, as shown in Figure 4, is composed by 200 rod beams with different cross section areas. A total number of 3 design variables are used to define the cross section area of the structural beams according to type and location as shown in Table 3. The grouping is carried out in order to make the optimization feasible, since an optimization of each single beam might not have been feasible.

The uncertainties considered in the numerical example are summarized in Table 2. Nodal loads are modelled as normal independently distributed variables. Each load corresponds to different physical actions applied to the structure, which are in order of increasing uncertainty respectively permanent, variable and natural actions (see Figure 4). The density of the material is also modelled as a random variable.

Table 2: Design variables and parameters of the steel Roof Truss.

Parameter	Description
Design Variable (A_1)	60 beams forming the top of the structure
Design Variable (A_2)	100 beams connecting the top and the bottom of the structure
Design Variable (A_3)	40 beams forming the bottom of the structure
maxDisp	Capacity of the system ($10^{-}3$ [m])
Parameter	Distribution(μ,σ)
Load (L_1)	normal (12000,120) [N]
Load (L_2)	normal (16000,800) [N]
Load (L_3)	normal (50000, 20000) [N]
Young's module (E)	lognormal(2.0e11, 1.0500e+10) [Pa]
Density (ρ)	normal (7500,150) [kg/m^3]

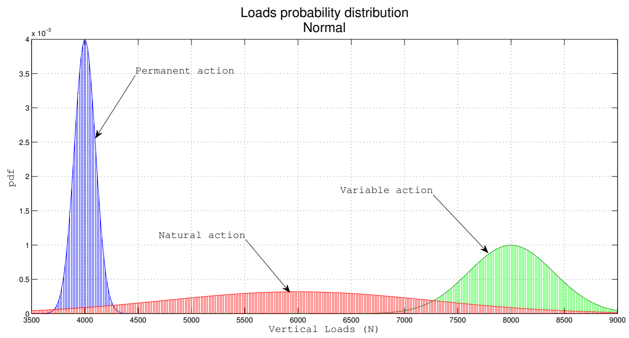

Figure 4: Distribution of the nodal loads.

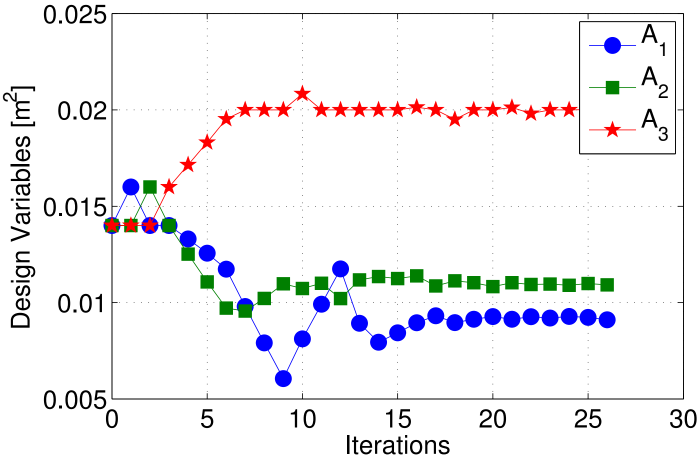

Figure 5: Evolution of the Design Variables (A_1, A_2 and A_3) during the reliability based optimization analysis.

Analysis The strategy adopted for the reliability based optimization of the steel roof truss is the following. The reliability based optimization analysis is performed adopting the so-called direct approach. The advanced Monte Carlo method namely Line Sampling is used to perform the reliability analysis at each iteration step of the optimization procedure. The line sampling allows to estimate the failure probability with only 60 model evaluations. The COBYLA algorithm is used to drive the optimization procedure.

The evolution of the design variables, objective function and the constraint during the optimization are shown in Figures 5,6 and 7, respectively.

The results of the analysis shows that a the total volume is decreased from the initial value of 6.3 to 5.7 [m^3]. The evolution of the design variables shows that the beam section A_3 is larger than the starting design while the design variables A_1 and A_2 was reduced. The failure probability of the system has been successfully reduced from an

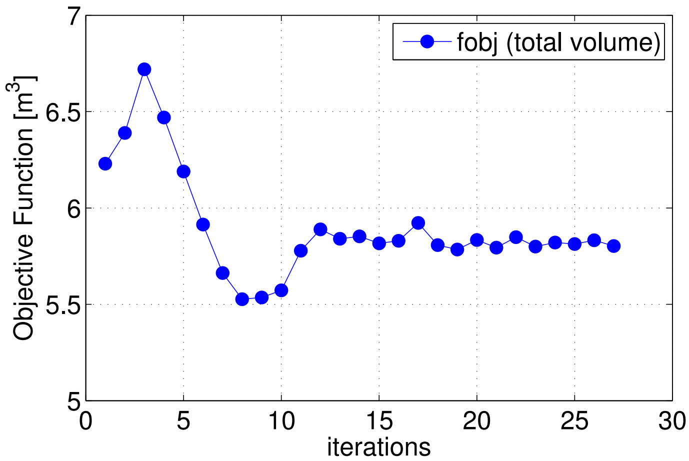

Figure 6: Evolution of the objective function (i.e. total volume) during the reliability based optimization analysis.

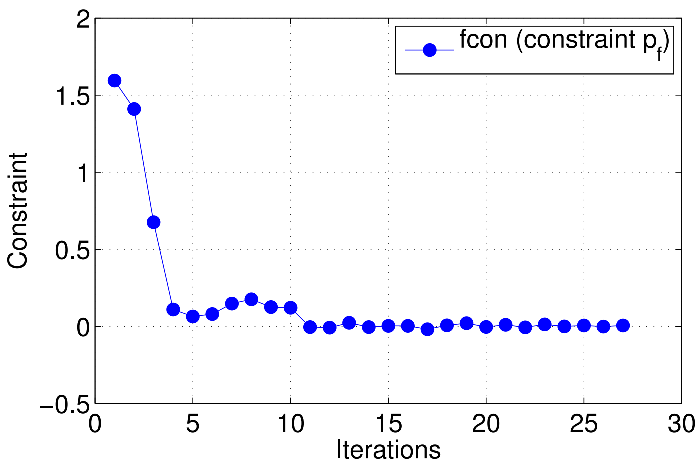

Figure 7: Evolution of the constraint (i.e. max admissible failure probability) during the reliability based optimization analysis.

initial value of $1.3 \cdot 10^{-2}$ to the prescribed value 10^{-4}.

It is important to notice that the total computational efforts required by the reliability based optimization analysis adopting the Line sampling and Cobyla (1200 model evaluations) represents only a small fraction of a single direct reliability analysis based on Monte Carlo simulation ($\approx 10^5$ model evaluations).

4.2 Truss Tower

Description of the problem In this second example a truss tower, subjected to dynamic random excitation, is analysed (see Figure 8). The purpose of the example is to minimize the total cost associate to the truss tower considering the effects of uncertainties. The costs associated to the truss tower are defined as the costs required by the material plus the costs associated to the unavailability of the service. More specifically, the material costs have been assumed to be proportional to the total volume of the beams multiplied by a unit cost, U_1. The costs associated to the unavailability of the service have been assumed to be proportional to the probability of failure of the tower by a unit cost U_2. Please note that in this numerical example the failure is related to the serviceability and not to the ultimate limit state.

Hence the object function has been defined as follow:

$$F(\cdot) = p_f * U_1 + V_{tot} * U_2 \tag{6}$$

In this numerical example no constraint functions are taken into account.

The truss tower is composed by twenty-five rod structural elements. These elements are assigned to three different groups (i.e. A_1, A_2 and A_3) that define different cross section areas. System failure occurs when the nodal displacements are beyond the allowed displacement, as shown in Figure 8. Structural response depends on ground acceleration, which is herein represented as a Gaussian stochastic process. The Youngs modulus and density of the material, attributed to each rod element, are modelled as normal independently distributed random variables.

Table 3: Design variables and parameter of the truss tower.

Parameter	Description
Design Variable (A_1)	[0.1 ·10^{-}3 -2 ·10^{-}3] [m^3]
Design Variable (A_2)	[0.1·10^{-}3 -4.6 ·10^{-}3] [m^3]
Design Variable (A_3)	[2·10^{-}3-4.6·10^{-}3]·10^{-}3 [m^3]
Unit Cost failure U_1	1000 [cost/(unavailability)]
Unit Cost material U_2	0.02 [cost/m^3]
maxDisp	Capacity of the system (0.024 [m])
Young's module (E)	normal($6.9 \cdot 10^{10}$, $3.4 \cdot 10^9$) [Pa]
Ground Acceleration (ρ)	Gaussian stochastic process (0,150) [m/s^2]

Random dynamic excitation is applied to the structure as time history acceleration by using stochastic process. Samples of the random excitation are generated using

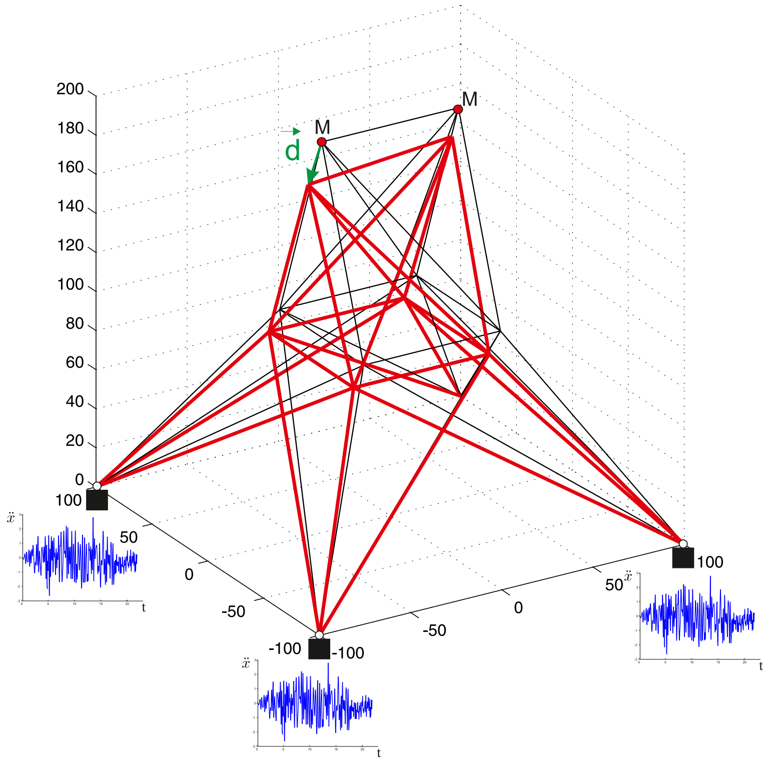

Figure 8: Truss Tower subject to random excitation.

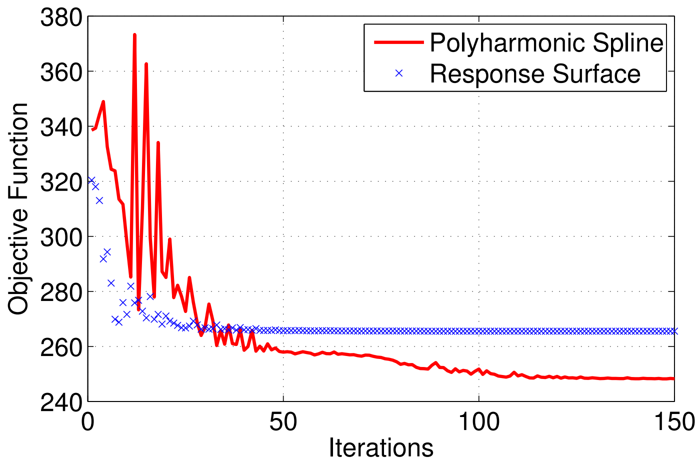

Figure 9: Evolution of the objective functions during the reliability based optimization analysis analysis adopting response surface and poly-harmonic spline.

Karhunen-Loève expansion. The covariance of the stochastic process is computed from an available set of experimental accelerograms, representing the ground acceleration during an earthquake.

Analysis Because of the computational costs associate to estimation of the failure probability of the system a direct approach is here not feasible. For this reason global meta-models have been used to approximate the failure probability of the system. Two different types of meta-model have been compared: response surface and polyharmonic spline. The training data required to calibrate the meta-model have been computed using Monte Carlo simulation (i.e. 27 full reliability analysis have been carried out) with 10^5 samples. The calibration of meta-model(s) is the most critical and the most computational demanding phase of the reliability based optimization analysis. In is important to mention that once the training data are available, more meta-models can be calibrated and used at practically no additional computational cost.

Then an unconstrained optimization, adopting the SIMPLEX algorithm, has been performed using the calibrate meta-models. Figures 9 and 10 show the evolution of the objective functions and the design variables during the optimization phase, respectively.Thanks to the surrogate models, the optimization problem can be performed also adopting a large number of iterations. In this example the Simplex algorithm has required up to 150 iterations in order to identify the optimum.

The results of this numerical example show that the meta-model allows to achieve a better reduction in term of the total costs (i.e. a lower value of the objective function). The main problem with the response surface is its inaccuracy to model extreme values of the failure probability (i.e. zeros and ones). In fact the response surface leads to inadmissible values of the failure probabilities while using the polyharmonic spline

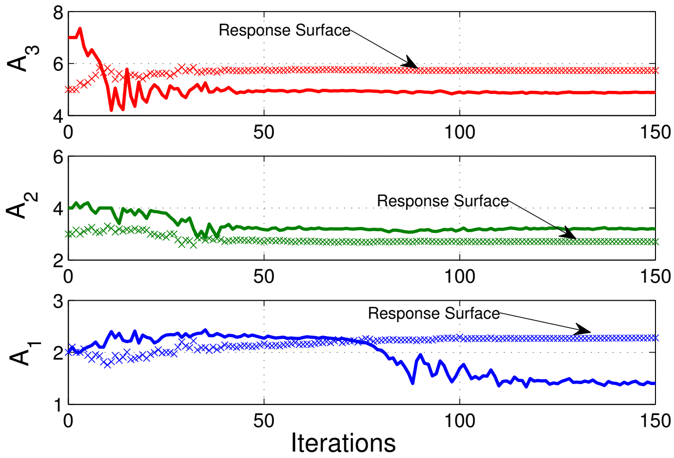

Figure 10: Evolution of the design variables during the reliability based optimization analysis adopting response surface and poly-harmonic spline.

the failure probability results correctly bounded between 0 and 1.

5 Conclusions

Combining optimization and reliability methods, it is possible to perform the so-called reliability based optimization analysis and robust design optimization [22], which seeks to identify optimal design solutions considering uncertainties and consequently the risk. Such types of analyses are an integral part of the development cycle to make the end-product, *i.e.* the component or system, less sensitive to factors that could adversely affect the performance.

In this work, a general purpose and flexible computational framework for solving reliability based optimization problem has been presented. The presented toolkit, namely OpenCossan, enables reliability based optimization analysis to be performed by combining a number of different strategies and methods that represent the state-of-the-art of the reliability and optimization algorithms. The computational framework developed is open, portable, flexible and extensible. It has been designed to be intuitive and user friendly allowing the users to apply very efficient tools and methods without an extensive training. Hence, it can certainty be useful to engineers and researchers, who are willing to develop in a non graphical mode, to test and develop new strategies and solution sequences. The applicability and the flexibility of the proposed framework for solving real-life problems have been demonstrated by means two applications considering static and dynamic load.

In conclusion, it is crucial that stochastic tools and procedures are offered to users for practical applications within easy-to-use general purpose software to strengthen the link between industries and academic researchers.

References

[1] Free software foundation, gnu lesser general public license, http://www.gnu.org/licenses/lgpl.html.

[2] J.A Anderson. *Introduction to Neural Network*. MIT Press, Cambridge, MA, 1995.

[3] J. Arora (Ed.). *Optimization of Structural and Mechanical Systems*. World Scientific, 2007.

[4] S.-K. Au and J.L. Beck. Estimation of small failure probabilities in high dimensions by subset simulation. *Probabilistic Engineering Mechanics*, 16(4):263–277, 2001.

[5] J. L. Beck, E. Chan, A. Irfanoglu, and C. Papadimitriou. Multi-criteria optimal structural design under uncertainty. *Earthquake Engineering & Structural Dynamics*, 28:741–761, 1999.

[6] G. E. P. Box and K.B. Wilson. On the experimental attainment of optimum conditions (with discussion). *Journal of the Royal Statistical Society Series B*, 13(1):1–45, 1951.

[7] Andrew R. Conn, Katya Scheinberg, and Luis N. Vicente. Review of surrogate model management, 2009.

[8] John O. Rawlings; Sastry G. Pantula; David A. Dickey. *Applied Regression Analysis*. Mathematics and Statistics. Springer New York, 1998.

[9] O. Ditlevsen and P. Bjerager. Methods of structural systems reliability. *Structural Safety*, 3(3-4):195–229, August 1986.

[10] O. Ditlevsen and O.H. Madsen. *Structural Reliability Methods*. John Wiley and Sons, 1996.

[11] J. Duchon. Splines minimizing rotation-invariant seminorms in sobolev spaces. *Constructive Theory of Functions of Several Variables, Lecture Notes in Mathematics*, 571:85–100, 1977.

[12] I. Enevoldsen and J.D. Sørensen. Reliability-based optimization in structural engineering. *Structural Safety*, 15(3):169–196, 1994.

[13] B. Goller, H.J. Pradlwarter, and G.I. Schuëller. Reliability assessment in structural dynamics. *Journal of Sound and Vibration*, 2011, submitted.

[14] Hector A. Jensen. Structural optimization of linear dynamical systems under stochastic excitation: a moving reliability database approach. *Computer Methods in Applied Mechanics and Engineering*, 194(12-16):1757–1778, April 2005.

[15] R. H. Myers and D. C. Montgomery. *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. John Wiley and Sons, Inc., New York, NY, second edition edition, 2002.

[16] E. Patelli, H.M.Panayirci, M. Broggi, B. Goller, P. Beaurepaire, H. J. Pradlwarter, and G. I. Schuëller. General purpose software for effficient uncertainty management of large finite element models. *Finite Elements in Analysis and Design*, 51:31–48, 2012.

[17] E. Patelli, H.J. Pradlwarter, and G.I. Schuëller. On multinormal integrals by importance sampling for parallel system reliability. *Structural Safety*, 33:1–7,

2011.
[18] R. Rackwitz. Reliability analysis - a review and some perspectives. *Structural Safety*, 23:365–395, 2001.
[19] G. I. Schuëller, M. Gasser, J. Hartl, and G. Lener. Practical design procedures using reliability based optimization. *Journal of Advanced Manufact. Systems*, 4(1):101–120, 2001.
[20] G. I. Schuëller, H. J. Pradlwarter, and P. S. Koutsourelakis. A critical appraisal of reliability estimation procedures for high dimensions. *Probabilistic Engineering Mechanics*, 19(4):463–474, 2004.
[21] G.I. Schuëller. Efficient Monte Carlo simulation procedures in structural uncertainty and reliability analysis - recent advances. *Journal of Structural Engineering and Mechanics*, 32(1):1–20, 2009.
[22] G.I. Schuëller and H.A. Jensen. Computational methods in optimization considering uncertainties - An Overview. *Computer Methods in Applied Mechanics and Engineering*, 198(1):2–13, 2008.
[23] G. Gary Wang and S. Shan. Review of metamodeling techniques in support of engineering design optimization. *Journal of Mechanical Design*, 129(4):370–381, 2007.