

COSSAN SOFTWARE: A MULTIDISCIPLINARY AND COLLABORATIVE SOFTWARE FOR UNCERTAINTY QUANTIFICATION

Edoardo Patelli¹, Matto Broggi², Silvia Tolo¹, Jonathan Sadeghi¹

¹Institute for Risk and Uncertainty, University of Liverpool (UK)
e-mail: edoardo.patelli@liverpool.ac.uk

² Institut für Risiko und Zuverlässigkeit, Leibniz Universität Hannover (DE)
e-mail: broggi@bauinf.uni-hannover.de

Keywords: COSSAN-X, OpenCOSSAN, Uncertainty Quantification, Risk Assessment

Abstract. *Computer-aided modelling and simulation is now widely recognised as the third ‘leg’ of scientific method, alongside theory and experimentation. Many phenomena can be studied only by using computational processes such as complex simulations or analysis of experimental data. In addition, in many engineering fields computational approaches and virtual prototypes are used to support and drive the design of new components, structures and systems.*

A general purpose software for uncertainty quantification and risk assessment, named COSSAN, is under continuous development. This is a multi-disciplinary software that satisfies industry requirements regarding numerical efficiency and analysis of detailed models that can be used to solve a wide range of industrial and scientific problems. The main aim of the COSSAN software is to allow the inclusion of non-deterministic analyses as a practice standard routing in scientific computing. This paper presents two recent toolboxes added to the OPENCOSSAN: Credal Networks and Interval Predictive model.

1 INTRODUCTION

Uncertainty quantification is a key requirement and challenge across various disciplines and industry is fully aware of the importance of non-deterministic analysis. This is particularly true in many engineering fields where computational approaches and virtual prototypes are used to predict and simulate complex systems. The continuous growth of computational power available allows to simulate the effect of the uncertainties on the model under analysis with increased precision. These advancements have allowed engineering practitioners to reduce the number of expensive experimental tests and analyse conditions that can not be reproduced.

However, in order to increase the confidence in the digital models, the effect of unavailable uncertainty needs to be simulate as well. In fact, improved model prediction can only be accomplished when different sources of uncertainty are explicitly included in the analysis [2]. Hence, uncertainty management is necessary to provide support to the decision makers. Unfortunately, industrial design methods are still predominantly deterministic. This is because of limited expertise in uncertainty quantification in industrial design offices together with lack of proper software for uncertainty quantification and stochastic analysis.

Software for uncertainty quantification have received considerably less attention than their deterministic counterparts mainly due to the computational costs associated with non-deterministic analysis.

In general, deterministic analysis provides a map between a single point in the input space (i.e. the model parameters) and a point in the output space (i.e. component or system performance). Stochastic analysis extends this map to a region in the input space and a corresponding region in the output space by repeating the deterministic analysis many times as represented is Figure 1. Generalised probabilistic approaches such as interval and fuzzy methods, imprecise probabilities and any combination thereof (see e.f. [3, 15]) allow to cope with model uncertainties, errors in modelling and measurements and noise in signals require in general the evaluation of different models stochastic models further increasing the computational cost of the analysis.

In order to increase the knowledge transfer between academia and industry and to include stochastic analysis as standard procedure in engineering practice, G.I Schuëller initiated the development of the COSSAN project at the Institute for Engineering Mechanics, University of Innsbruck, Austria [?]. The COSSAN project aims at developing a new generation of a general purpose software for non-deterministic analysis that can be used by industry, academics, researchers and for teaching purpose as well. The software incorporates the knowledge, understanding and intellectual property from more than 30 years of research in the field of computational stochastic analysis.

Starting from 2006, the next generation software referred as COSSAN-X is under continuous development and it is intended for a wider range of applications in different fields, which includes optimization analysis, life-cycle management, reliability and risk analysis, sensitivity, optimization and robust design [17]. Since 2012, an open source version of COSSAN-X, called OPENCOSSAN, is available under the Lesser GNU licence [9]. This means that the program can be used for free, redistributed and modified under the terms of the GNU General Public License. The OPENCOSSAN aims to promote learning and understanding of non-deterministic analysis through the distribution of an intuitive, flexible, powerful and open computational toolbox in Matlab environment [16]. The combination of various algorithms with specific solution sequences permits the analysis of different engineering and scientific problems.

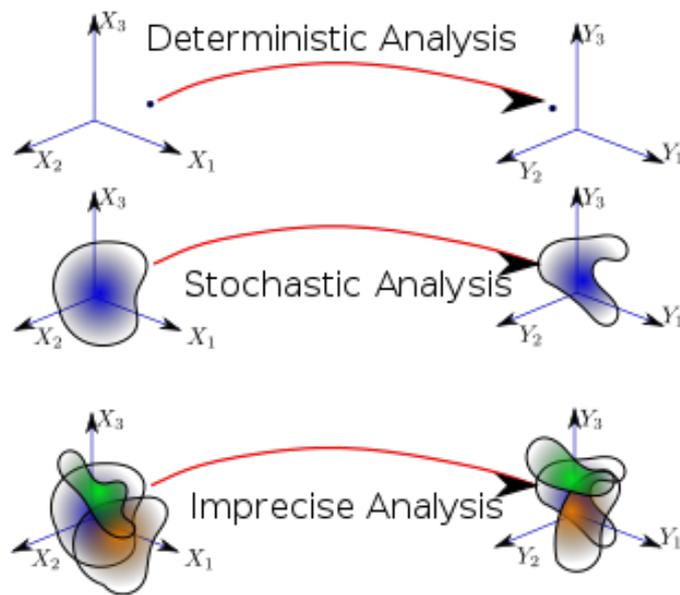


Figure 1: Mapping between the input space and the output space in case of deterministic analysis, stochastic analysis and generalised (imprecise) analysis, respectively.

1.1 Software developing approach

Nowaday, COSSAN software is developed collaboratively between the Liverpool Institute for Risk and Uncertainty and the Institut für Risiko und Zuverlässigkeit from the Leibniz Universität Hannover (DE) and under continuous improvements [14, 16].

OPENCOSAN, the computational core of the COSSAN software, is released under the terms of the GNU Lesser General Public License [9]. The source code is accessible via available at the web address <http://www.cossan.co.uk>. The Apache Subversion system is used for software versioning and revision control system distributed to keep tracks of changes and contribution from different developers. OPENCOSAN, is coded exploiting the object-oriented Matlab programming environment, where it is possible to define specialized solution sequences, which include reliability methods, sensitivity analysis, optimization strategies, surrogate models and parallel computing strategies. The computational framework is organized in packages. A package is a namespace for organizing classes and interfaces in a logical manner, which makes large software project OPENCOSAN easier to manage. A class describes a set of objects with common characteristics such as data structures and methods. Objects, that are instances of classes can be aggregated forming more complex objects and proving solutions for practical problem in a compact, organized and manageable format. OPENCOSAN provides intuitive, clear, well documented and human readable interfaces to the classes. No acronyms are used to define methods and properties.

The availability of an extended documentation, tutorials and examples is of a key importance for the usability of a software. The COSSAN documentation is written collaboratively by COSSAN developers and end-users using MediaWiki tool where with rare exceptions, articles can be edited by anyone.

COSSAN software is developed as a general purpose software which provides generic functionalities and can be changed and expanded through user-defined routines and additional codes.

This implies that solutions for solving very specific problems are not generally available although they can be added by the users. Instead, methods and tools that can be used for solving a reasonably wide range of engineering and scientific problems are prioritised. One of the challenges developing COSSAN software is maintaining its flexibility and modularity as well its usability for the end-user. Another key element of the software is a comprehensive risk management and uncertainty quantification based on different representation of the uncertainties based on probabilistic approaches, interval and fuzzy methods, imprecise probabilities and any combination thereof.

In the following sections two different toolboxes recently introduced into OPENCOSAN are shown.

2 Meta Models toolbox

As shown in Figure 1, stochastic analysis requires the repeated evaluation of a (detailed) numerical model. The analysis time can be reduced significantly by using meta-models, which approximate the quantities of interest at low computational costs. In other words, meta-models mimic the behaviour of the original model (e.g. FE model), by means of a mathematical model with negligible computational cost.

3 Interval Predictor Models toolbox

Interval Predictor Models (IPMs) are a class of meta-model which describe the expected spread of the output of a model whilst making very few assumptions about the model. The meta-model finds an upper and lower bound for the predicted output of the model using a function chosen by the user (at present limited to radial or polynomial basis). Therefore, once trained, the meta-model can make predictions very quickly [6].

Interval Predictor Models offer a robust quantification of uncertainty, even when few data points are available. This makes them useful when a very computationally expensive model limits the number of simulations which can be performed. The robust quantification of uncertainty is achieved in two ways. Firstly, a measure of uncertainty is intrinsic to the meta-model since the output is an interval - its spread describes our uncertainty. Secondly, the reliability, R , of this interval i.e. the probability future data points will fall inside the interval is bounded by

$$Prob_{P^n}[R \geq R^*] \geq C, \quad (1)$$

where R^* and C are the bounds on the reliability of the model and our confidence in it respectively [4]. R^* as a function of C is calculated by OPENCOSAN. For this model it is recommended to use these quantities as a method of measuring calibration rather than the coefficient of determination which is normally used in OPENCOSAN. In addition, the samples in the training data set must be independently distributed [5].

3.1 Comparison to other meta-models

In one sense Interval Predictor Models are similar to Gaussian Processes. However, Interval Predictor Models are useful in situations where Gaussian Processes may be too computationally expensive, for example where there are many data points and therefore it is too computationally expensive to make predictions. In addition, Interval Predictor Models have the advantage that they often yield tighter bounds than Gaussian Processes.

The Interval Predictor model requires more time to train than a response surface, however it offers a very robust quantification in the epistemic uncertainty resulting from the construction of the meta-model.

The amount of data points which can be used to create the IPM is limited only by the time the user is prepared to spend on the training the meta-model, which is performed by convex optimisation. In fact, for fewer than tens of thousands of data points the time required to train the model is in general negligible.

3.2 Work flow and implementation details

The IPMs implemented into `OPENCROSSAN` are the linear parameter dependency type-1 IPMs described in [6].

The IPM class is a subclass of the meta-model abstract class in `OPENCROSSAN`, and therefore inherits most of its behaviour in a similar way to the response surface class. However some of the arguments which must be specified to the constructor method are crucially different, and therefore they are described below:

- `type`: a string containing radial or polynomial, which specifies which basis type should be used.
- `maximumexponent`: a positive integer giving the maximum index of polynomial to be used in a full factorial polynomial basis. If the basis is polynomial and this is not specified then exponents must be specified.
- `exponents`: a matrix of indices - one row for each monomial, one column for each input variable. Mandatory for polynomial basis unless `maximumexponent` is specified.
- `augmented`: An integer (0, 1 or 2) specifying if the IPM is augmented (see [6] for explanation). Optional (default to 0).
- `sigma`: the sigma matrix for radial basis. Mandatory when type is radial.
- `centers`: the centres matrix for radial basis. Mandatory when type is radial.

Once training is complete the IPM parameters are stored in the matrices `MIPMParametersU` and `MIPMParametersL`.

The implemented meta-model class IPM provides a method (`removeOutliers`) to remove the outlier which implements the outlier removal algorithm described in [6]. In simple terms, this method will remove training data which prevents the spread of the IPM being reduced, and then re-calibrate the IPM.

The performance of the IPM model can be used to calculate using the `reliabilityPlot` method and finally, for models with 1 or 2 inputs it is possible to visualise the IPM using the `plotModel` method.

3.3 Application

In order to demonstrate the application of the class, an IPM surrogate of the Rosenbrock function is constructed. The created IPM has a polynomial basis of degree 2. Figure 2 shows a plot of R^* as a function of C . If 20 more samples are taken from the function 96.25% fall inside the IPM which is in agreement with Fig. 2. In order to create a more robust IPM than this either the degree of the polynomial used must be decreased or more training data must be obtained.

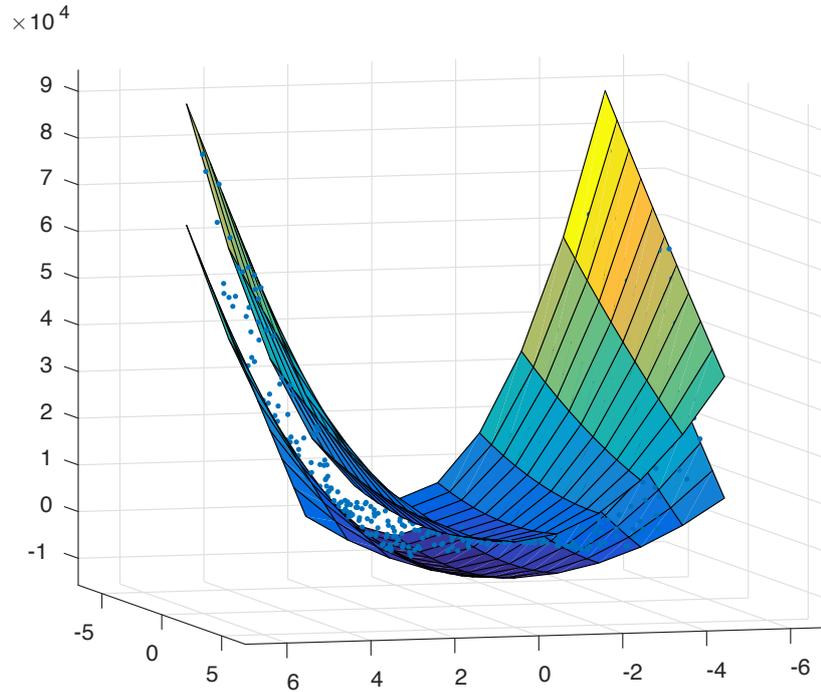


Figure 2: Degree 2 polynomial type-1 IPM trained on 200 samples from Rosenbrock function.

Now the user may choose to use the output from the meta-model in their performance function. For example if the output represented a quantity which must be below a certain value (such as displacement of a cantilever beam) then the user would choose the upper bound of the displacement from the IPM and then calculate an upper bound on the failure probability. This could be repeated with the lower bound in order to calculate a lower bound on the failure probability.

4 Credal Networks toolbox

4.1 General remarks

Bayesian Belief Networks, more commonly known as Bayesian Networks (BNs), are a probabilistic graphical model based on the use of directed acyclic graphs, integrating graph theory with the robustness of Bayesian statistics. The graphical framework of such models consists of nodes, representing the variables of the problem of interest, connected to each other by edges, generally arrows, that depict the dependency link existing between two nodes. Each variable is assumed conditionally independent of its non-descendants given its parent variables: this propriety is known as local Markov propriety. Each node of the network is linked to a Conditional Probability Distribution (CPD) that, according to the local Markov propriety, defines the strength of the probabilistic dependency existing between each individual node and its parents. When a node has no parents, i.e. it is a *root* of the network, a marginal probability distribution is associated with it.

The main aim of the Bayesian Network approach is to factorize the probability of a complex event exploiting the knowledge regarding the dependencies existing among its sub-parts. BNs are extremely attractive for a large variety of applications providing decomposition procedure and the possibility to update the overall knowledge regarding the event of interest when new

information (i.e. *evidence*) about any of its sub-elements becomes available.

BNs exact inference algorithms (i.e. adopting an analytical approach) are almost exclusively limited to the use of crisp probability values which are not able to fully capture the data generally available. This often leads to the adoption of discretization procedure which empoverish the quality of the information available. On the other hand, approximate inference algorithms allow the use of continuous probabilistic variables but at the cost of lower accuracy, and they can result often inefficient or have unknown rates of convergence. The limitations of the approximate inference approach result even more significant when considering the field of risk analysis and decision making, or more generally those applications where rare events and near-real-time computation play a crucial role [?].

Generalizing further, even traditional continuous probabilistic variables are often not able to fully capture the uncertainty of data without introducing strong subjective assumptions not justified by the information available. Alternatively, imprecise probabilities theory provides several mathematical frameworks able to depict the uncertainty inevitably affecting the data and hence reducing the introduction of subjective biases [1].

In order to overcome the limitations associated with traditional BNs, the integration of such approach with the imprecise probability theory has attracted increasing attention in the scientific community leading to the formulation and study of Credal Networks (CNs). CNs can be considered as a more general approach that relaxes the requirements of BNs. They still rely on the use of directed acyclic graphs but differently from the former method allow the use of imprecise probabilities as input parameters of the model, ensuring a higher degree of flexibility in the representation of the data uncertainty. In spite of the capabilities and promising potential of this methodology the application of such models is still quite limited, mainly due to the high computational cost associated with CNs inference.

4.2 Work flow and implementation

Further efforts and research are strongly required in order to enhance the attractiveness of CNs outside the academic world and to ensure the reliability and efficiency of their performance in real-world applications. Well known and novel methodologies are integrated in the software in order to provide the implementation, manipulation and analysis of CNs. The main tasks of the toolbox can be listed as:

- Implementation of the CN model defined by the user
- Reduction of the initial user-defined model (to ease the burden of the inference computation)
- Inference computation of the network
- Sensitivity analysis of the reduced model

The following sections offer a brief introduction to each of the mentioned tasks.

Implementation The toolbox provides the definition of initial models allowing a range of possible variables (and subsequently node types) such as:

- events with a discrete and finite number of outcome states associated with crisp probability values (*discrete* nodes)

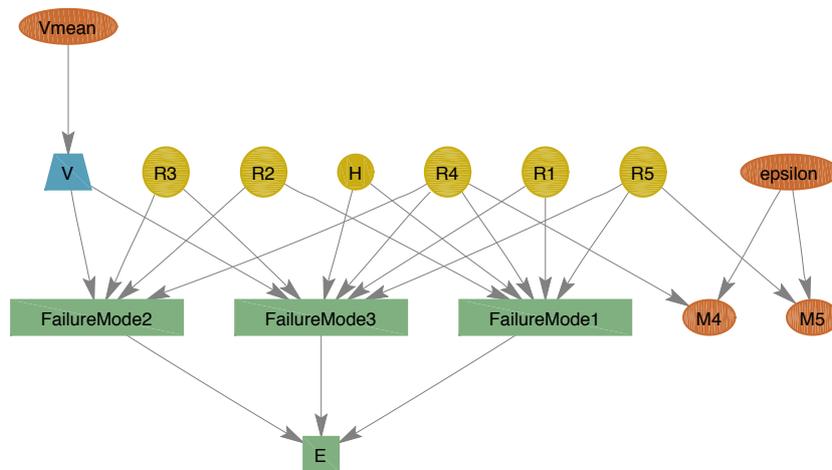


Figure 3: Example of the directed graph of a simple structural system subject to vertical load V and horizontal load H

- events with a discrete and finite number of outcome states associated with interval probability values (*discrete nodes*)
- events whose outcome states are described by continuous probabilistic distributions (*probabilistic nodes*)
- events whose outcome states are described by interval variables (*bounded nodes*)
- events whose outcome states are described by imprecise probability distributions (*hybrid nodes*)

The example in Figure 3 refers to an elementary structural system [11, 8]) consisting of one-bay elastoplastic frame subject to a vertical load V and horizontal load H . The structure is defined by the plastic moment capacities R_1, R_2, R_3, R_4, R_5 , and presents three possible failure modes ($FailureMode1, FailureMode2, FailureMode3$ in Fig.3) defined by as many limit-state functions. The occurrence of any of these failure modes is assumed to result in the failure of the whole system (event E in the network). The nodes M_4 and M_5 refer to measurement of the plastic capacities R_4 and R_5 and are affected by a measurement error $epsilon$. The plastic moment capacities are assumed to have a probabilistic distributions, while the measurements M_4 and M_5 , being affected by the error $epsilon$ represented by an interval variable, are bounded nodes (as indeed $epsilon$). The vertical load is supposed to have a probabilistic distribution whose mean is not precisely known but represented by an interval: the load V is then represented as a *hybrid* node differently from the load H assumed as probabilistic.

In Fig.3 the rectangular shaped nodes are discrete, the oval shaped ones bounded, the circular ones probabilistic and finally the trapezoidal shaped nodes are hybrid. The graphical representation of the network is realized using the Bioinformatics toolbox for MATLAB [12] and it is designed in order to highlight the different nature of the model variables.

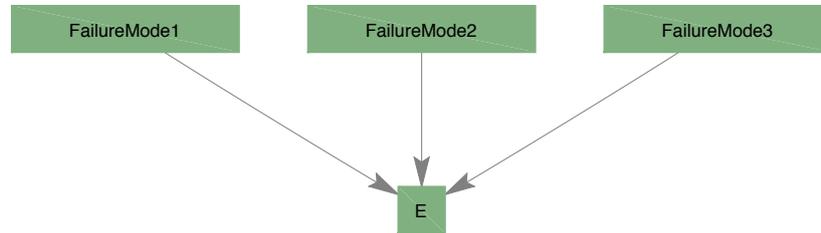


Figure 4: Final network resulting from the reduction of the model in Fig.3

Reduction A key procedure of the methodology implemented in the toolbox consists of reducing the initial model to an equivalent one containing only discrete nodes. The implemented reduction procedure ensures the preservation of the initial information. For instance, the children nodes *FailureMode1*, *FailureMode2* and *FailureMode3* can be expressed by the related limit-state functions, which are defined in function of the parents nodes $R_1, R_2, R_3, R_4, R_5, H$ and V . The computation of the probability bounds is obtained adopting numerical methods for structural reliability analysis. A range of options are available in the toolbox. The default option relies on the use of the Advanced Line Sampling technique [?].

The two main phases of the reduction procedure are:

- Computation of continuous nodes children of at least one continuous node.
When also continuous nodes are expressed as domains in the outcome space of their parents they are automatically computed according to their nature:
 - Probabilistic nodes: probabilistic distributions are constructed over the samples collected from the parent nodes through the use of Monte Carlo methods.
 - Bounded nodes: the bound are obtained through a random search in the outcome space of their parents.
 - Hybrid nodes: the current implementation is limited to the use of parametric imprecise random variables. Conversely, the imprecise parameters of the distribution can be expressed as parent nodes of the hybrid ones, as for the nodes V_{mean} and V in Figure 3.
- Computation of discrete nodes children of at least one continuous node.
According to the nature of the nodes involved (this time the parent nodes) the option available are different:
 - Only probabilistic and discrete parent nodes: in this case the approach reduces to the Enhanced Bayesian Network methodology [18] and a wide variety of method-

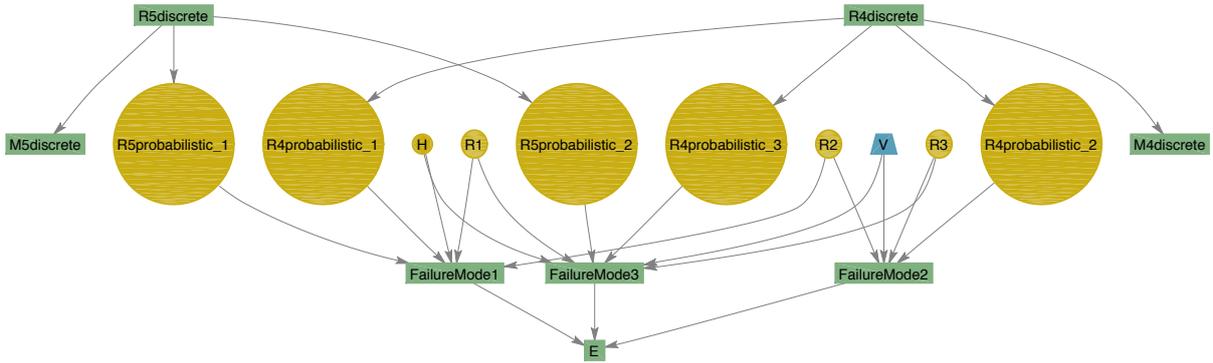


Figure 5: Network resulting from the model in Fig.3 after the computation and discretisation of the initial bounded nodes

ologies can be adopted (e.g. FORM, SORM, traditional and Advanced Monte Carlo methods etc.)

- Bounded, probabilistic and discrete parent nodes: two options are available, one consisting of a generalized version of FORM based on the use of convex sets [10], one on the Advanced Line Sampling methods. It is opportune that only this latter method allows to compute probability bounds on the event of interest while the formers one focuses only on the estimation of the worst-case scenario probability.
- Hybrid, bounded, probabilistic or discrete parent nodes: currently the only option available for the computation is the Advanced Line Sampling [?].

The result of the reduction of the model of Fig.3 is shown graphically in Fig.4. As shown by the graph, the nodes M_4 and M_5 have been removed from the reduced network, since non-discrete and not receiving any evidence. In order to process evidence, it is necessary that the node receiving it is present in the reduced model. Subsequently, if a non-discrete node receives evidence, it is necessary to discretize it in order to keep it in the final model. In this case, the first phase of the reduction procedure, i.e. the computation of continuous nodes children of at least one continuous node, is not entirely carried out but the discretization of the continuous nodes receiving evidence is automatically performed, avoiding their elimination. Let assume that some measurement regarding the plastic moments capacities R_4 and R_5 is available and hence the nodes M_4 and M_5 receive evidence in the network. In this case, the modified network containing the discretized information of M_4 and M_5 is shown in Fig.5. The procedure followed for the discretization is similar to that proposed by Straub and Kiureghian [18] and aims to limit the inevitable loss of information quality associated with such procedure. The final network obtained from the reduction of the model in Fig.5 is shown in Fig.6: in this latter the nodes $M_{4discrete}$ and $M_{5discrete}$ refer to the measurements of the plastic moment capacity and can hence receive evidence updating the overall estimate associated with the probability of failure of the structure.

Inference Computation Three inference computational methods are available in the toolbox:

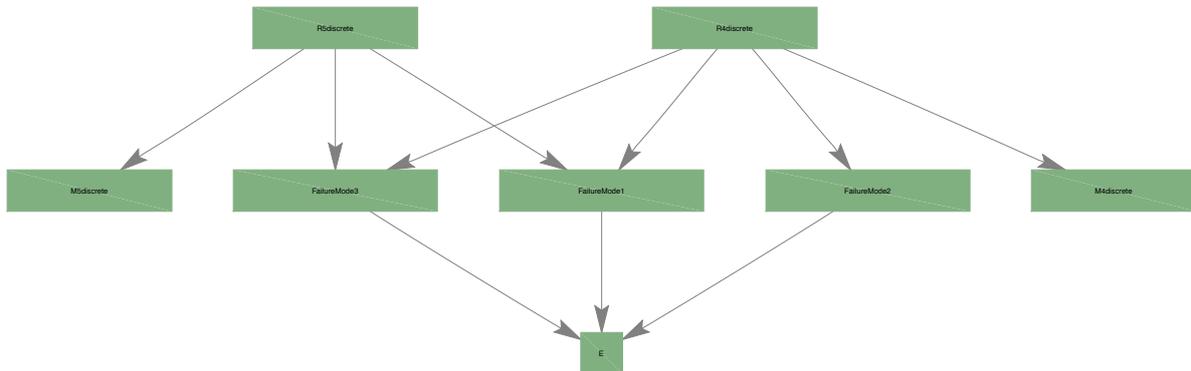


Figure 6: Final network resulting from the reduction of the model in Fig.5

- **Coarse Approximation:** the lowest computational cost associated with the inference computation, implying on the other hand a lower level of accuracy providing an outer approximation of the true value of the output.
- **Fine Approximation:** based on a original and novel inference computational algorithm which aims to achieve a high degree of accuracy at a lower computational cost than that associated with exact inference. The method provides the true value of the query probability bounds when no evidence is introduced in the network and provides inner approximation values otherwise [19].
- **Exact Inference:** currently this option is restricted to the adoption of a combinatorial approach to inference computation. This can be performed either using the built in variable elimination algorithm.

Sensitivity Analysis A novel method allows to identify the possible changes of input parameters (i.e. interval conditional probabilities associated with the nodes of the network) that result in the achievement of the user-defined degree of accuracy in output (i.e. probability bounds of the query probability). In the current implementation the method is limited to the use of boolean nodes and relies on a combinatorial approach. The results provide an intuitive understanding of when parameter changes do or do not matter in terms of query robustness. Further detail can be found in [20].

5 Conclusion

The availability of an intuitive and easy to use general purpose software is a key for making the non-deterministic analysis a common practice in computational models and numerical simulations. In fact, uncertainty and imprecision are unavoidable and they must be accounted for in any analyses. Only implementing stochastic analysis, digital (or virtual) design will be credible and applicable in different sectors and fields. Digital design allows to design fast and better, reducing the design costs and provide cost-effective and feasible engineering solutions.

COSSAN-X provides an easy to use yet powerful software for uncertainty quantification and risk management that can be used by analysis without an extensive learning curve and allowing to create a bridge between the academic research and the industrial practice. The engine of the COSSAN software OPENCOSAN is freely available making the software development more sustainable, continuously updated and avoiding code duplication. It is an excellent collaborative tools for researchers and academics encourages the cross-discipline utilisation of stochastic analysis.

REFERENCES

- [1] M. Beer, S. Ferson, and V. Kreinovich. Imprecise probabilities in engineering analyses. *Mechanical Systems and Signal Processing*, 37(12):4 – 29, 2013.
- [2] Michael Beer, Kok Kwang Phoon, and Ser Tong Quek, editors. *Special Issue: Modeling and Analysis of Rare and Imprecise Information*, volume 32 of *Structural Safety*. 2010.
- [3] Michael Beer, Yi Zhang, Ser Tong Quek, and Kok Kwang Phoon. Reliability analysis with scarce information: Comparing alternative approaches in a geotechnical engineering context. *Structural Safety*, 41(0):1 – 10, 2013.
- [4] Marco C. Campi and Simone Garatti. A sampling-and-discarding approach to chance-constrained optimization: feasibility and optimality. *Journal of Optimization Theory and Applications*, 148(2):257–280, 2011.
- [5] M.C. Campi, G. Calafiore, and S. Garatti. Interval predictor models: Identification and reliability. *Automatica*, 45(2):382 – 392, 2009.
- [6] Luis G Crespo, Sean P Kenny, and Daniel P Giesy. Interval predictor models with a linear parameter dependency. *Journal of Verification, Validation and Uncertainty Quantification*, 1(1), 2016.
- [7] Marco de Angelis, Edoardo Patelli, and Michael Beer. Advanced line sampling for efficient robust reliability analysis. *Structural Safety*, 52:170–182, 2015.
- [8] Armen Der Kiureghian et al. First-and second-order reliability methods. *Engineering design reliability handbook*, pages 14–1, 2005.
- [9] Free Software Foundation. Free software foundation, gnu lesser general public license, version 3, 2007.
- [10] Yangjun Luo, Zhan Kang, and Alex Li. Structural reliability assessment based on probability and convex set mixed model. *Computers & Structures*, 87(21):1408–1415, 2009.
- [11] HO Madsen, S Krenk, and NC Lind. *Structural method of safety*, 1986.
- [12] MathWorks. MATLAB [computer software]. Natick, MA.
- [13] Kevin Murphy et al. The bayes net toolbox for matlab. *Computing science and statistics*, 33(2):1024–1034, 2001.

- [14] Edoardo Patelli. *Handbook of Uncertainty Quantification*, chapter COSSAN: A Multi-disciplinary Software Suite for Uncertainty Quantification and Risk Management, pages 1–69. Springer International Publishing, Cham, 2016.
- [15] Edoardo Patelli and Matteo Broggi. On general purpose software for the efficient uncertainty management of large finite element models. In *NAFEMS World Congress 2013 9-12 June 2013, Salzburg, Austria*. NAFEMS, 9-12 June 2013 2013.
- [16] Edoardo Patelli, Matteo Broggi, de Angelis Marco, and Michael Beer. OpenCossan: An efficient open tool for dealing with epistemic and aleatory uncertainties. In *Vulnerability, Uncertainty, and Risk*, pages 2564–2573. American Society of Civil Engineers, June 2014.
- [17] Edoardo Patelli, Hussain Murat Panayirci, Matteo Broggi, Barabara Goller, Pierre Beaurepaire Helmut J. Pradlwarter, and Gerhart I. Schuëller. General purpose software for efficient uncertainty management of large finite element models. *Finite Elements in Analysis and Design*, 51:31–48, 2012.
- [18] D. Straub and Der Kiureghian. Bayesian network enhanced with structural reliability methods: methodology. *Journal of engineering mechanics*, 136(10):12481258, 2010.
- [19] S. Tolo, E. Patelli, and M. Beer. An inference method for Bayesian networks with probability intervals. *Structural Safety*, Submitted on July 2016.
- [20] S. Tolo, E. Patelli, and M. Beer. Sensitivity analysis for Bayesian networks with interval probabilities. In *Proceedings of the 26th European Safety and Reliability Conference, ESREL, 25th 29th September 2016*, Glasgow, U.K., to be published.
- [21] Silvia Tolo, Edoardo Patelli, and Michael Beer. Robust vulnerability analysis of nuclear facilities subject to external hazards. *Stochastic Environmental Research and Risk Assessment*, pages 1–24.