Safety and Reliability: Methodology and Applications – Nowakowski et al. (Eds)
© 2015 Taylor & Francis Group, London, ISBN 978-1-138-02681-0

Safe reliability assessment through probabilistic constraint reasoning

Elsa Carvalho, Jorge Cruz & Pedro Barahona
Centro de Inteligência Artificial, Universidade Nova de Lisboa, Portugal

ABSTRACT: Reliability quantifies the ability of a system to perform its required function under variable conditions. The adequate functioning of a system is often represented by inequality constraints, and its reliability is the probability that such constraints are satisfied, given the uncertainty the variables are subject to. Since this computation is very hard, namely when the systems are modeled with nonlinear constraints, traditional methods adopt a number of approximations, thus computing a value that may be far from the exact one. Moreover, these methods do not provide any guarantees regarding the correctness of the computed results. In this paper, we use the probabilistic continuous constraints framework to efficiently compute safe bounds for the reliability of a system, and illustrate it on a number of representative examples.

1 INTRODUCTION

Reliability analysis studies the ability of a system to perform its required function under variable conditions. In this context, reliability assessment quantifies the chance of system failure at any stage of a system's life. This research area has a wide range of applications, including the aeronautical (Nikolaidis, Ghiocel, & Singhal 2007), chemical (Goel, Grievink, Herder, & Weijnen 2002) and building (Huang, Chan, & Lou 2012) industries.

Reliability is reported in terms of the probability of adequate functioning of a system and its exact quantification requires the calculation of a multi-dimensional integral with a nonlinear integration boundary. Because there is rarely a closeform solution, this calculation is one of the major concerns of classical approaches to solve reliability problems, which adopt approximation methods that rely on several simplifications of the original problem to compute a reliability estimate, often leading to inaccurate results, especially in highly nonlinear problems.

When choosing among several alternative configurations of a system it is important to obtain safe bounds for their reliability, since designs with a high reliability estimate but high uncertainty on such estimation are not credible options. Such safe bounds are not available with classical approaches, but they can be provided by the Probabilistic Continuous Constraint framework that we have been developing to integrate probabilities and constraint programming, and that has already been successfully applied to inverse problems (Carvalho, Cruz, & Barahona 2013).

The paper is organized as follows. Section 2 briefly introduces the concepts of reliability assessment and describes classical techniques, such as FORM (Hohenbichler & Rackwitz 1983), SORM (Fiessler, Neumann, & Rackwitz 1979), and Monte Carlo simulation (Halder & Mahadevan 1999), used to address such problem, discussing their limitations. In sections 3 and 4 we introduce our Probabilistic Continuous Constraint approach, discussing the features that are required toaddress reliability assessment. Section 5 illustrates our framework on a set of representative examples, comparing it with traditional methods, highlighting the importance of the safe bounds we efficiently obtain. The last sectionsummarizes the main conclusions and directions for future work.

2 RELIABILITY ASSESSMENT

A limit-state is a condition beyond which a system no longer fulfills the desired functionality. Reliability assessment calculates and predicts the probability of limit-state violations at any stage of a system's life.

The probability of occurrence of a limit-state violation in a system represents its probability of failure, P_f, whereas $P_s = 1 - P_f$ represents its reliability.

Failure events are represented as limit-state constraints g is a limit-state function and x is a realization of the random vector X (defined in Ω_X) that represents all the relevant uncertainties influencing the probability of failure and has joint Probability Density Function (PDF) f_X. Thus,

the failure event is $F = \{x \in \Omega_{\boldsymbol{X}} : g(x) < 0\}$ and the probability of failure is the probability of the failure event:

$$P_f = P(x \in F) = \int_{g(x)<0} f_{\boldsymbol{X}}(x)dx \tag{1}$$

More generally, we are interested in problems that can be defined by one or more limit-state functions. In series systems, global failure occurs when at least one limit-state functions is violated, whereas in parallel systems, it occurs when all limit-state functions are violated. So, the failure events for series and parallel systems with k limit-state functions, are, respectively:

$$F = \bigcup_{i=1}^{k} \{x \in \Omega_{\boldsymbol{X}} : g_i(x) < 0\} \qquad \text{and} \tag{2}$$

$$F = \bigcap_{i=1}^{k} \{x \in \Omega_{\boldsymbol{X}} : g_i(x) < 0\} \tag{3}$$

2.1 *Classical techniques*

Reliability assessment involves the calculation of a multi dimensional integral in a possibly highly nonlinear integration boundary (Equation (1)). Analytical computation of such integral is usually impossible, so various simulation-based and numerical methods have been proposed to deal with this problem.

In (Hasofer & Lind 1974), Hasofer and Lind introduced the reliability index technique for calculating approximations of the desired integral with reduced computation costs. The reliability index has been extensively used in the first and second order reliability methods (FORM (Hohenbichler & Rackwitz 1983) and SORM (Fiessler, Neumann, & Rackwitz 1979)).

The main idea is to move the reliability problem from the space of random vector $\boldsymbol{X}$ to the space of standard normal statistically independent random variables $\boldsymbol{U} = \langle U_1, \ldots, U_n \rangle$ using a suitable transformation $\boldsymbol{U} = T(X)$, such as Rosenblatt (Rosenblatt 1952) or Nataf (Nataf 1962) transformations (see (Hohenbichler & Rackwitz 1981, Melchers 1999) for an overview). In the $\boldsymbol{U}$ space, Equation (1) can be expressed as:

$$P_f = \int_{g(\boldsymbol{u})<0} f_{\boldsymbol{U}}(\boldsymbol{u})d\boldsymbol{u} = \prod_{i=1}^{n} \int_{g(u)<0} \phi_{U_i}(u_i)du_i$$

where ϕ_{U_i} is the standard normal PDF of U_i.

In FORM an approximation to the probability of failure is obtained by making the failure surface, $g(\boldsymbol{U}) = 0$, linear at the design point, $\boldsymbol{u}^*$, often called the Most Probable Point of failure (MPP). This is the point on the failure surface closest to the origin and with the highest probability (local maximum) in the failure domain of the standard normal space. The distance from the origin to the design point is the reliability index $\beta = \|\boldsymbol{u}^*\|$.

Since the standard normal space is rotational symmetric, probability of failure can be directly obtained using the reliability index, $P_f = \Phi(-\beta)$, where Φ is the standard normal cumulative probability function.

As the limit state function is in general nonlinear it is not possible to know the design point in advance and this has to be found iteratively. The design point is thus, the solution to the constrained optimization problem:

$$\beta = \min_{u \in \{g(u)=0\}} \|u\|$$

This problem, being the most expensive part of FORM algorithm, may be solved in a number of different ways (see (Eldred, Bichon, & Adams 2006) for an overview). An appropriate iteration scheme converges after some iterations, providing the design point $\boldsymbol{u}^*$ as well as the reliability index β, which may be related directly to the probability of failure. However, with non convex optimization problems, it is not guaranteed that the solution point will be the global minimum-distance point.

FORM usually works well when the failure surface has only one minimal distance point and the function is nearly linear in the MPP neighborhood. However, for increasingly nonlinear failure surface the probability of failure estimated by FORM becomes increasingly inaccurate (and possibly unreasonable) (Melchers 1999). To address such non linearity SORM incorporates some curvature in the limit state approximation through a parabolic approximation to the failure surface (Breitung 1984).

In both methods, it is assumed a single limit-state function with a single design point where only the region around such point contributes to the probability of failure. In limit-state functions with multiple design points (and in problems with multiple limit-state functions), application of FORM or SORM around a single design point results in erroneous estimates for the probability of failure. So, the problem of identifying the multiple design points was addressed by several authors (e.g. (Kiureghian & Dakessian 1998, Barranco-Cicilia, de Lima, & Sudati-Sagrilo 2009)).

In series systems, generally there exists one design point for each limit-state function (which contributes to the identification of the feasible region). In contrast, in parallel systems there usually exists one design point for each pairwise inter-

section of limit-state functions (again contributing to the identification of the unfeasible region).

Once all design points are identified, FORM or SORM approximations are constructed at these points and the failure probability is computed by series system reliability analysis (for multiple design points in a single limit-state function or series systems) or by parallel system reliability analysis (for parallel systems) (see (Sørensen 2004, Notes 6 and 7) for details).

Other techniques include sampling, based on Monte Carlo simulation (MCS) (Halder & Mahadevan 1999) and work well for small reliability requirements. Nevertheless, as the desired reliability increases, the number of samples must also increase to find at least one infeasible solution.

Since Monte Carlo method is basically a sampling process, the results are subjected to sampling error that decreases with the sample size. However, using procedures known as variance reduction techniques the error may be reduced without significantly increasing the sample size. One of such procedures with a high convergence rate is the Monte Carlo with Importance Sampling (MCIS) (Melchers 1999). In MCIS, the regions of interest for the simulation process are those around the points in the failure domain having the largest values, i.e., the design points.

Given the simplifications adopted and their approximate nature, none of the above methods provides guarantees on the reliability values computed, specially for nonlinear problems. In contrast, the Probabilistic Continuous Constraint framework does not suffer from this limitation, guaranteeing safe bounds for the probability of failure.

3 CONTINUOUS CONSTRAINT PROGRAMMING

Continuous constraint programming has been widely used to model safe reasoning in applications where uncertainty arises. A Continuous Constraint Satisfaction Problem (CCSP) (Lhomme 1993, Benhamou, McAllester, & van Hentenryck 1994, Sam-Haroud & Faltings 1996) is a triple $\langle X, D, C \rangle$ where X is a tuple of n real variables $\langle x_1, \ldots, x_n \rangle$, D is a Cartesian productof intervals $I_1 \times \ldots \times I_n$ (a box), where each I_i is the domain of variable x_i and C is a set of numerical constraints (equations or inequalities) on subsets of the variables in X. A solution of the CCSP is a value assignment toall variables satisfying all the constraints in C. The feasible space $\mathcal{F}$ is the set of all CCSP solutions within D.

Constraint reasoning relies on branch-and-prune algorithms to obtain sets of boxes that cover exact solutions for the constraints (the feasible space). These algorithms begin with an initial crude cover of the feasible space (the initial search space, D) which is recursively refined by interleaving pruning and branching steps until a stopping criterion is satisfied. The branching step splits a box from the covering into sub-boxes (usually two). The pruning step either eliminates a box from the covering or reduces it into a smaller (or equal) box maintaining all the exact solutions. Pruning is achieved through a constraint propagation algorithm ($\mathcal{CPA}$) which combines constraint propagation and consistency techniques (Benhamou, Goualard, Granvilliers, & Puget 1999) based on interval analysis methods (Moore 1966).

Interval analysis is an extension of real analysis that allows computations with intervals of reals instead of reals. Common operations and unary functions are extended for interval operands. For instance, [1,2] + [3,6] results in the interval [4,8], which encloses all the results from a point-wise evaluation of the real arithmetic operator on all the values of the operands. In practice these extensions simply consider the bounds of the operands to compute the bounds of the result, since the involved operations are monotonic.

Interval extensions [f], allow computing enclosures of the range of real functions f, over boxes. The natural interval extension of a function is obtained by evaluating its expression for interval arguments using interval arithmetic.

The branch-and-prune algorithm usually maintains two coverings of the feasible space $\mathcal{F}$ of a CCSP $\langle X, D, C \rangle$: one outer and one inner box cover, as follows.

Outer Box Cover A set of almost disjoint boxes[1] $\mathcal{F}_{\square} = \{B_1, \ldots, B_n\}$, where $\forall_{1 \le i \le n} (B_i \subseteq D \wedge vol(B_i) > 0)$, is an outer box cover of $\mathcal{F}$ iff $\mathcal{F} \subseteq \bigcup_{i=1}^{n} B_i$.

A complementary concept is that of inner box cover, where an inner box of a CCSP is a box totally contained in the feasible space.

Inner Box A box $B \subseteq D$ with $vol(B) > 0$ is an inner box of $\mathcal{F}$ iff $B \subseteq \mathcal{F}$.

Inner Box Cover A set of almost disjoint boxes $\mathcal{F}_{\blacksquare} = \{B_1, \ldots, B_n\}$ is an inner box cover of $\mathcal{F}$ iff $\bigcup_{i=1}^{n} B_i \subseteq \mathcal{F}$.

We are particularly interested in maintaining an inner box cover that is a subset of the outer box cover.

[1] Two boxes B_1 and B_2 are almost disjoint iff $vol(B_1 \cap B_2) = 0$. The volume of a box $B \subseteq \mathbb{R}^n$ is the product of the width of its intervals, i.e., $vol(B) = wid(I_1) \times \ldots \times wid(I_n)$.

Joint Box Cover A joint box cover $\mathcal{F}_{\boxplus}$ of $\mathcal{F}$ is a pair $\langle \mathcal{F}_{\square}, \mathcal{F}_{\blacksquare} \rangle$, with $\mathcal{F}_{\blacksquare} \subseteq \mathcal{F}_{\square}$.

Often it is important to know whether a joint box cover represents the feasible space more accurately than another. For this purpose we define a partial order ($\preceq$) as follows.

Tighter Joint Box Cover Given a CCSP $\langle X, D, C \rangle$ and two joint box covers of $\mathcal{F}$, $\mathcal{F}_{\boxplus 1} = \langle \mathcal{F}_{\square 1}, \mathcal{F}_{\blacksquare 1} \rangle$ and $\mathcal{F}_{\boxplus 2} = \langle \mathcal{F}_{\square 2}, \mathcal{F}_{\blacksquare 2} \rangle$, $\mathcal{F}_{\boxplus 2}$ is a tighter joint box cover of $\mathcal{F}$ than $\mathcal{F}_{\boxplus 1}$, (written $\mathcal{F}_{\boxplus 2} \preceq \mathcal{F}_{\boxplus 1}$) iff:

$$\bigcup \mathcal{F}_{\square 1} \supseteq \bigcup \mathcal{F}_{\square 2} \text{ and } \bigcup \mathcal{F}_{\blacksquare 1} \subseteq \bigcup \mathcal{F}_{\blacksquare 2}$$

The branch-and-prune algorithm *B & P* (Algorithm 1, similar to (Granvilliers & Benhamou 2006)) receives a joint box cover and computes a tighter one. For that purpose, and while the stoping criterion is not satified, the algorithm removes a box from the outer box cover that, simultaneously, verifies the *eligible* predicate and is selected by the *order* function (line 2), and splits it (line 3). Subsequently it modifies the inner and outer box covers of the joint cover. If the retrieved box is already in the inner box cover (line 4) then it is replaced by the boxes resulting from the *split*, which are also added to the outer box cover. Otherwise (line 5) the boxes resulting from the *split*, are pruned by the constraint propagation algorithm and added to the outer box cover (line 6). Those that are inner boxes are also added to the inner box cover (line 7).

Typically, in Algorithm 1, the *split* function splits the box by the midpoint of its largest interval; the *inner* predicate relies on natural interval extensions

```
   Input: ⟨F□, F■⟩: joint box cover; C: set of
          constraints; ε: real; stop, order:
          function;
   Output: ⟨F□, F■⟩out: joint box cover;
1  while (¬stop(⟨F□, F■⟩)) do
2      B ← remove(F□, eligible, order);
3      if B = ∅ then break;
4      S ← split(B);
5      if B ∈ F■ then
6          F■ ← F■ \ {B};
7          L■ ← S;   L□ ← S;
8      else
9          L□ ← {CPA(Bi, C) : Bi ∈ S};
10         L■ ← {Bi ∈ L□ : inner(Bi, C)};
11     end
12     ⟨F□, F■⟩ ← ⟨F□ ∪ L□, F■ ∪ L■⟩;
13 end
14 return ⟨F□, F■⟩;
```

Algorithm 1. *B & P* ($\langle \mathcal{F}_{\square}, \mathcal{F}_{\blacksquare} \rangle$, C, ε, *stop*, *order*).

induced by each constraint[2], replacing the variables by the intervals of the box, and checking whether all values in the resulting interval are solutions for that constraint; and the $eligible_{\varepsilon}$ predicate requires the width of the largest interval of the box to be larger than a given ε. The *stop* predicate imposes the stopping criterion and the *order* function specifies which box is selected for processing. Both *stop*, *order* and ε are parameterizable.

4 PROBABILISTIC CONSTRAINT PROGRAMMING

In classical CCSPs, uncertainty is modeled by intervals that represent the domains of the variables. Constraint reasoning reduces uncertainty providing a safe method for computing a set of boxes enclosing the feasible space. Nevertheless this paradigm cannot distinguish between different scenarios and all combination of values within such enclosure are considered equally plausible. In (Carvalho 2012) and (Carvalho, Cruz, & Barahona 2013) the authors proposed the Probabilistic Continuous Constraint paradigm (hereafter referred as PC), that extends the continuous constraint framework with probabilistic reasoning, allowing to further characterize uncertainty with probability distributions over the domains of the variables.

4.1 *Probabilistic continuous constraints*

Probability provides a classical model for dealing with uncertainty (Halpern 2003). The basic elements of probability theory are a) random variables and b) events, which are appropriate[3] subsets of the sample space Ω. A probabilistic model is an encoding of probabilistic information that allows the probability of events to be computed, according to the axioms of probability. In the continuous case, the usual method for specifying a probabilistic model assumes a full joint PDF over the considered random variables.

In order to complement the interval bounded representation of uncertainty with a probabilistic characterization of the values distributions, we define a probabilistic continuous constraint space. Firstly, a probability space is associated with a CCSP.

PC Probability Space Given a CCSP $\langle X, D, C \rangle$, the associated probability space is $\langle \Omega, \mathcal{B}^n, P \rangle$ where $\Omega \supseteq D$, $\mathcal{B}^n$ is the n-dimensional Borel σ-algebra on Ω and P is a probability measure.

Secondly, the variables of the CCSP are mapped onto random variables.

[2] In this paper we consider only inequality constraints.
[3] In the sense that it is possible to assign them a measure.

PC Random Vector Given a PC probability space on $\langle\langle x_1, \ldots, x_n\rangle, D, C\rangle$, an identity random vector $\mathbf{X} = \langle X_1, \ldots, X_n\rangle$ is considered, such that $X_i : \mathbb{R}^n \to \mathbb{R}$ is defined as $X_i(\Omega) = \Omega_i$.

Thirdly, the probability measure P is defined.

PC Probability Measure Given a PC probability space, PC random vector $\mathbf{X}$ with joint PDF $f(\cdot)$ and an event $\mathcal{H} \in \mathcal{B}^n$, the probability measure P is defined as:

$$P(\mathcal{H}) = \int \cdots \int_{\mathcal{H}} f(x_1, \ldots, x_n) dx_n \ldots dx_1$$

Finally, a probabilistic continuous constraint problem space may be defined.

PC Problem Space A PC problem space is a pair $\langle\langle X, D, C\rangle, f\rangle$ where $\langle X, D, C\rangle$ is a CCSP and f is the joint PDF of the PC random vector $\mathbf{X}$.

In this context, a PC event, hereafter referred as $\mathcal{H}$, is the feasible space of a CCSP $\langle X, D, C\rangle$.

4.2 *Multidimensional integration over a region*

In general the multidimensional integral to obtain the probability of a PC event cannot be easily computed, since it may have no closed-form solution or the event may establish a complex nonlinear integration boundary. The PC framework relies on continuous constraint programming to get a tight joint box cover $\langle \mathcal{H}_{\square}, \mathcal{H}_{\blacksquare} \rangle$ of the region of integration $\mathcal{H}$ and on Taylor models integration techniques to compute safe enclosures for the integrals over the obtained boxes.

Taylor Models A Taylor model of $g : \mathbb{R}^n \to \mathbb{R}$ inside an n-dimensional box B is a pair $\langle p, R\rangle$, where p is a polynomial and R is an interval satisfying, $\forall \boldsymbol{x} \in B$, $g(x) \in p(x) + R$. The order of the Taylor model is the degree of p.

A Taylor model of a function can be obtained from its multivariate Taylor expansion, using the interval evaluation of the highest order derivatives to compute rigorous bounds for the remainder.

Lemma 4.1 (From (Berz & Makino 1999)) *Given a Taylor model $\langle p, R\rangle$ of a function $g : \mathbb{R}^n \to \mathbb{R}$ inside an n-dimensional box B*:

$$[I]_{TM}(B, g) = \int_B p(x)dx + R\, vol(B) \ni \int_B g(x)dx$$

$[I]_{TM}$ can be used to obtain a sharp enclosure for the integral of a function over some region defined as a box (for more details about verified quadrature with Taylor models see (Goldsztejn, Cruz, & Carvalho 2014)).

Input: $\langle\langle X, D, C\rangle, f\rangle$: PC problem space; ε, δ: double
Output: I: interval;
1 $stop_\delta(\mathcal{H}_{\boxplus}) \equiv wid([P_{\mathcal{H}}](\mathcal{H}_{\boxplus}, f)) \leq \delta$;
2 $\mathcal{H}_{\boxplus} \leftarrow B\&P(\langle\{D\}, \varnothing\rangle, C, \varepsilon, stop_\delta, order_P)$;
3 **return** $[P_{\mathcal{H}}](\mathcal{H}_{\boxplus}, f)$;

Algorithm 2. *probEnclose*($\langle\langle X, D, C\rangle, f\rangle, \varepsilon, \delta$).

4.3 *Probability of a PC event*

The probability of a PC event can be enclosed by summing up the integral contributions of all the boxes from its joint cover, as follows.

Enclosure for the Probability of $\mathcal{H}$ Given a joint box cover $\mathcal{H}_{\boxplus} = \langle \mathcal{H}_{\square}, \mathcal{H}_{\blacksquare} \rangle$ of a PC event $\mathcal{H}$, an enclosure for the probability of $\mathcal{H}$ is given by[4]:

$$[P_{\mathcal{H}}](\mathcal{H}_{\boxplus}, f) = \sum_{B_i \in \mathcal{H}_{\blacksquare}} [I]_{TM}(B_i, f) + \sum_{B_i \in \mathcal{H}_{\square} \setminus \mathcal{H}_{\blacksquare}} [0] \uplus [I]_{TM}(B_i, f)$$

Notice that for non inner boxes, where the feasible region is some unknown subset of the box, the integral ranges from zero to the integral of the function over the box. In this case it is no longer worth computing a sharp (and more costly) enclosure and a lower order Taylor model can be used.

Since Taylor Models are used to compute safe enclosures for the integral over each box, the result is guaranteed to include the correct probability value.

Algorithm 2 computes bounds for the probability of a PC event. It uses the *B & P* algorithm where the $stop_\delta$ predicate imposes a specified accuracy δ for the probability enclosure computed over its joint box cover argument (line 1) and the $order_P$ function specifies that the box with highest uncertainty in its probability enclosure is chosen first.

For a PC problem space, with an associated PC event $\mathcal{H}$, *B & P* computes increasingly tighter covers of $\mathcal{H}$ until the intended accuracy δ for its probability is reached (line 2). The final joint box cover is used to compute an enclosure for the probability of $\mathcal{H}$[5] (line 3). The parametrization of *B & P* implies choosing boxes with higher uncertainty in their probability, in order to reduce such uncertainty. In practice, the accumulation of round-

[4]The union hull interval operator $\uplus$ returns the smallest interval containing both interval arguments.

[5]In fact, in the implementation of the algorithm, the probability enclosure is maintained and updated during the process, to check the stop criterion, and then returned in the end.

ing errors may prevent the algorithm to deliver the required accuracy. When this accuracy is too sharp, *B & P* may stop because there are no more eligible boxes (all the boxes are already small wrt ε) without achieving the required accuracy.

4.4 *Reliability assessment*

Reliability analysis can be used to analyze existing systems, thus being a significant support for those in charge of decision-making. In the following we describe how the PC framework can be used to obtain safe results on such problems.

For the formulation of a reliability assessment problem as a PC problem space, we distinguish between series and parallel systems.

Parallel and Series Systems as a PC problem space Consider a parallel system with an associated random vector $\boldsymbol{X} = \langle X_1, \ldots, X_n \rangle$ with joint PDF f_x defined in $\Omega_{\boldsymbol{X}} \subseteq \mathbb{R}^n$ and a set of k limit-state functions that define the failure event F as in (3). This system is modeled as a PC problem space, $\langle\langle X, D, C\rangle, f\rangle$, such that:

$$D \subseteq \Omega_{\boldsymbol{X}} \qquad X = \langle x_1, \ldots, x_n \rangle$$
$$C = \{g_i(x) \leq 0 : 1 \leq i \leq k\} \qquad f = f_{\boldsymbol{X}}(x)$$

Its probability of failure is $P_f = P(\mathcal{F}(\langle X, D, C\rangle))$ and its reliability is $P_s = 1 - P_f$.

The formulation of a series system as a PC problem space is adapted from the previous, where $C = \{g_i(x) \geq 0 : 1 \leq i \leq k\}$ and $\mathcal{F}(\langle X, D, C\rangle)$ defines the success event. So the computed probability $P_s = P(\mathcal{F}(\langle X, D, C\rangle))$ is the reliability of the system.

From these formulations, Algorithm 2 can be used to compute the probability of event $\mathcal{H} = \mathcal{F}(\langle X, D, C\rangle)$ and obtain enclosures for the reliability (or probability of failure) of series or parallel systems.

Notice that reliability problems do not impose bounds on the random variables, which is not possible to model in the PC framework, where $D \subseteq \Omega_{\mathbf{X}}$ must be a bounded box. Thus, to guarantee the safety of the computed probability enclosures when $D \subset \Omega_{\mathbf{X}}$, a small correction term must be added to such enclosure. This is done by computing $[P](\mathcal{D})$, an enclosure for the probability of event $\mathcal{D} = \mathcal{F}(\langle X, D, \{\}\rangle)$ and $[P](\boldsymbol{\Omega}_{\mathbf{X}} \backslash \mathcal{D}) = 1 - [P](\mathcal{D})$, an enclosure for the neglected probability. Then the term $[0] \uplus [P](\Omega_{\mathbf{X}} \backslash \mathcal{D})$ is added to the enclosure computed by Algorithm 2.

5 EXPERIMENTAL RESULTS

To illustrate the limitations of the classical techniques (FORM, SORM and Monte Carlo) described in Section 2.1, several examples of reliability assessment problems found in the literature are modeled as PC problem spaces. The algorithms were implemented over RealPaver 1.0 (Granvilliers & Benhamou 2006), and the experiments were carried out on an Intel Core i7 CPU at 2.4 GHz.

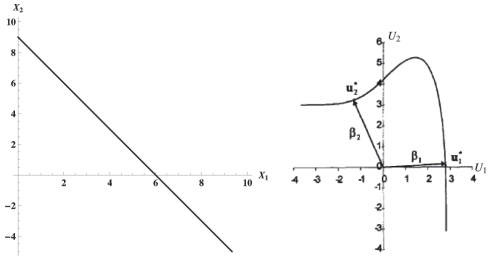
The results obtained with the classical approaches are compared with those computed with algorithm 2 with $\delta = 10^{-6}$, $\varepsilon = 10^{-15}$ and a Taylor order of 6 for inner boxes (and 1 otherwise), hereafter referred as PCTM. All the results obtained with the PCTM algorithm are presented after adding the correction term.

In the experiments, function *NProbability* (from *Mathematica* v9.0.0.0 Wolfram Research 2012)) with the default parametrization, is also used to compute the required probabilities as a complementary source of comparison. Although, in the proposed examples, all *Mathematica* estimates are within the enclosure computed by the PCTM algorithm, this is not the case in general (see (Carvalho, Cruz, & Barahona 2013) for some examples). Moreover different parameterizations provide different values. Although the differences might be small, *Mathematica* does not provide bounds for the errors.

The first example illustrates the non linearity induced in the limit-state function resulting from the transformation of a non Gaussian distribution into a standard normal distribution.

Example 5.1. Consider the reliability problem, originally introduced in Hohenbichler & Rackwitz 1981, with $\boldsymbol{X} = \langle X_1, X_2 \rangle$, joint PDF $f_{\mathbf{X}}(x_1, x_2) = (x_1 + x_2 + x_1 x_2) e^{-(x_1 + x_2 + x_1 x_2)}$ defined in $\Omega = [0, \infty] \times [0, \infty]$ and limit-state function $g(x_1, x_2) = 18 - 3x_1 - 2x_2$.

Although the limit-state function is linear in the original space, it becomes highly nonlinear and has two design points in the standard normal space, due to the strong non normality of the random variables. Figure 1 shows the limit-state function



(a) Linear limit-state. (b) $\boldsymbol{U} = T_1(\boldsymbol{X})$.

Figure 1. Linear limit-state in the original space and nonlinear in the standard normal space.

(a) in the original space and (b) in the standard normal space.

The problem is formulated as a PC problem space:

$$X = \langle x_1, x_2 \rangle \qquad D = [0,30] \times [0,30]$$
$$C = \{g(x_1, x_2) \leq 0\} \qquad f = f_{\boldsymbol{X}}(x_1, x_2)$$

The bounds for D guarantee a negligible probability for the neglected Ω region, $[P](\Omega \backslash \mathcal{D}) \leq 4.5 \times 10^{-11}$.

The results obtained with the classical approaches (from) and with the PCTM algorithm are shown in Table 1. It presents the approximations obtained by FORM and SORM methods when only one of the design points is considered and when both are considered for both transformations $T_1(\mathbf{X})$ and $T_2(\mathbf{X})$, described in (Hohenbichler & Rackwitz 1981). For Monte Carlo (MC) and PCTM algorithms this does not apply.

It is clear from Table 1 that the results obtained with FORM and SORM have a great variability, depending on the chosen configuration, and, except for one, are outside the safe enclosure computed by PCTM algorithm (in 0.12 seconds CPU time). Using *Mathematica* the obtained result (in 0.12 seconds CPU time) was 0.294486×10^{-2}.

The next example illustrates a nonlinear limit-state function where the original space is normal (although not standard normal).

Example 5.2. Consider the reliability problem from (Choi, Grandhi, & Canfield 2010), with $\boldsymbol{X} = \langle X_1, X_2 \rangle$, where $X_1 \sim \mathcal{N}(10,5)$ and $X_2 \sim \mathcal{N}(10,5)$ are independent random variables defined in $\Omega = \mathbb{R}^2$, and limit-state function $g(x_1, x_2) = x_1^4 + 2x_2^4 - 20$. The problem is formulated as a PC problem space:

$$X = \langle x_1, x_2 \rangle \qquad D = [-40,60] \times [-40,60]$$
$$C = \{g(x_1, x_2) \leq 0\}$$
$$f = \frac{1}{50\pi} e^{-\frac{1}{2}\left[\left(\frac{x_1-10}{5}\right)^2 + \left(\frac{x_2-10}{5}\right)^2\right]}$$

The bounds for D guarantee a negligible probability for the neglected Ω region, $[P](\Omega \backslash \mathcal{D}) \leq 2.8 \times 10^{-13}$.

The results obtained with the classical approaches (from (Choi, Grandhi, & Canfield 2010, pag. 132–136)) where two SORM versions are considered (see (Choi, Grandhi, & Canfield 2010, Chapter 4) for details) and with the PCTM algorithm are shown in Table 2.

The result obtained with FORM grossly overestimates the probability of failure. Those obtained with both versions of SORM are closer to the correct value, however are still far from the exact value. Simulation with Monte Carlo produces the result closer to the correct value with an 5.23% error. Using *Mathematica* the obtained result (in 1.17 seconds CPU time) was 0.185252×10^{-2}, which is within the enclosure computed by the PCTM algorithm (in 0.84 seconds CPU time).

The problems in the following example are found in (Sørensen 2004) to illustrate series and parallel systems.

Example 5.3. Consider the reliability assessment problems from (Sørensen 2004, Note 6) (series system) and (Sørensen 2004, Note 7) (parallel system), with $\boldsymbol{X} = \langle X_1, X_2 \rangle$, where X_1 and X_2 are independent standard normal random variables defined in $\Omega = \mathbb{R}^2$, and the limit state functions of Table 3.

Table 1. Probability of failure $\times 10^2$.

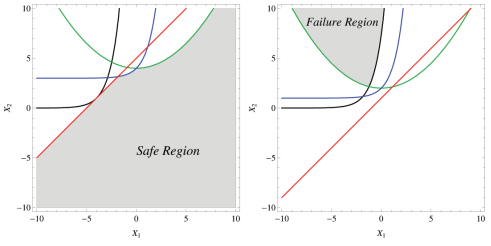
	u_1^* alone		u_2^* alone		u_1^* and u_2^*			
	FORM	SORM	FORM	SORM	FORM	SORM	MC	PCTM
$T_1(X)$	0.269	0.279	0.023	0.016	0.292	0.296		
$T_2(X)$	0.404	0.294	0.014	0.015	0.417	0.308	0.294	[0.294429, 0.294530]

Table 2. Probability of failure $\times 10^2$.

FORM	0.9005
SORM Breitung	0.2221
SORM Tvedt	0.2087
MC	0.1950
PCTM	[0.185162, 0.185263]

Table 3. Limit state functions.

	Series system	Parallel system
$g_1(x_1, x_2)$	$e^{x1} - x_2 + 3$	$e^{x1} - x_2 + 1$
$g_2(x_1, x_2)$	$x_1 - x_2 + 5$	$x_1 - x_2 + 1$
$g_3(x_1, x_2)$	$e^{x1+4} - x_2$	$e^{x1+2} - x_2$
$g_4(x_1, x_2)$	$0.1x_1^2 - x_2 + 4$	$0.1x_1^2 - x_2 + 2$



(a) Series system. (b) Parallel system.

Figure 2. Series and parallel systems.

Table 4. Probability of failure $\times 10^2$.

	Series system	Parallel system
SB	[0.02241, 0.05190]	[0.0762, 2.1692]
DB	[0.03516, 0.04091]	[0.1264, 0.2256]
PCTM	[0.031265, 0.031366]	[0.170314, 0.170415]

Figure 2 shows the limit-state functions and the safe/failure regions of these problems. The problems are formulated as PC problem spaces:

$$D = [-10,10] \times [-10,10] \qquad X = \langle x_1, x_2 \rangle$$

$$f = \frac{1}{2\pi} e^{-\frac{1}{2}(x_1^2 + x_2^2)}$$

$$C = \{g_i(x,y) \geq 0 : 1 \leq i \leq 4\} \qquad \text{series system}$$

$$C = \{g_i(x,y) \leq 0 : 1 \leq i \leq 4\} \qquad \text{parallel system}$$

In both problems 10 standard deviations around the mean value are assumed for the bounds of D, with $[P](\Omega \backslash \mathcal{D}) \leq 2.8 \times 10^{-13}$.

The results obtained with the classical approach for series and parallel systems analysis, where simple (SB) and Ditlevsen (DB) bounds are considered (see (Sørensen 2004, Notes 6 and 7) for details) and with the PCTM algorithm[6] are shown in Table 4.

We conclude that the simple bounds are too wide to be informative. The more accurate Ditlevsen bounds, for the series system, do not include the exact value in the safe enclosure computed by the PCTM algorithm (in 0.38 seconds CPU time). For the parallel system, they do include the safe enclosure computed by the PCTM algorithm (in 0.73 seconds CPU time) but are much wider. The results obtained with *Mathematica* were, for the series system, 0.0312962×10^{-2} (in 0.92 seconds CPU time) and, for the parallel system, 0.170394×10^{-2} (in 0.47 seconds CPU time).

[6]For the series system, the probability of failure was obtained from the reliability computed by the PCTM algorithm.

6 CONCLUSIONS AND FUTURE WORK

In this paper we propose to use the Probabilistic Continuous Constraints framework to deal with reliability assessment problems. Given its grounding on continuous constraint solving, this framework computes safe bounds for the reliability of series and parallel systems, contrary to classical approaches. The various kinds of approximations used by these approaches may turn the computed reliability value of little practical use, since they do not provide any bounds to the errors incurred. This is particularly significant in systems modeled by means of nonlinear constraints.

Moreover, the proposed framework, while guaranteeing the robustness of the computed values, does it very efficiently, being highly competitive when compared with *Mathematica*, that computes non-guaranteed results.

In the future the authors aim to extend the framework to address systems that are formulated as a combination of series and parallel components. Another interesting add-on would be the ability to model problems with a mixture of integer and continuous random variables, since these are an important class of problems appearing in science and engineering.

REFERENCES

Barranco-Cicilia, F., E.C.-P. de Lima, & L. Sudati-Sagrilo (2009). Structural Reliability Analysis of Limit State Functions With Multiple Design Points Using Evolutionary Strategies. *Ing. invest. y tecnol. 10*, 87–97.

Benhamou, F., F. Goualard, L. Granvilliers, & J. Puget (1999). Revising hull and box consistency. In *Procs. Int. Conf. on Logic Programming*, pp. 230–244. MIT Press.

Benhamou, F., D. McAllester, & P. van Hentenryck (1994). CLP (intervals) revisited. In *ISLP*, pp. 124–138. MIT Press.

Berz, M. & K. Makino (1999). New methods for high-dimensional verified quadrature. *Reliable Computing 5*, 13–22.

Breitung, K. (1984). Asymptotic Approximation for Multinormal Integrals. *J. Eng. Mech. 110*(3), 357–366.

Carvalho, E. (2012). *Probabilistic Constraint Reasoning*. Ph. D. thesis, Universidade Nova de Lisboa, Faculdade de Ciênciase Tecnologia.

Carvalho, E., J. Cruz, & P. Barahona (2013). Probabilistic constraints for nonlinear inverse problems. *Constraints 18*(3), 344–376.

Choi, S., R. Grandhi, & R. Canfield (2010). *Reliability-Based Structural Design*. Springer.

Eldred, M., B. Bichon, & B. Adams (2006). Overview of Reliability Analysis and Design Capabilities in DAKOTA. In *Workshop on Reliable Engineering Computing*, pp. 1–26.

Fiessler, B., H.-J. Neumann, & R. Rackwitz (1979). Quadratic limit states in structural reliability. *J. Engrg. Mech. Div. 105*, 661–676.

Goel, H., J. Grievink, P. Herder, & M. Weijnen (2002). Integrating reliability optimization into chemical process synthesis. Reliability Eng. and System Safety 78(3), 247–258.

Goldsztejn, A., J. Cruz, & E. Carvalho (2014). Convergence analysis and adaptive strategy for the certified quadrature over a set defined by inequalities. *J. of Comp. and Applied Math. 260*(0), 543–560.

Granvilliers, L. & F. Benhamou (2006). Realpaver: an interval solver using constraint satisfaction techniques. *ACM Trans. Math. Softw.* 32(1), 138–156.

Halder, A. & S. Mahadevan (1999). Probability, Reliability and Statistical Methods in Engineering Design. Wiley.

Halpern, J.Y. (2003). *Reasoning about Uncertainty*. MIT Press. Hasofer, A.M. & N.C. Lind (1974). Exact and invariant second-moment code format. *J. Engrg. Mech. Div.*

Hohenbichler, M. & R. Rackwitz (1981). Non-normal dependent vectors in structural safety. *J. Engrg. Mech. Div. 107*, 1227–1238.

Hohenbichler, M. & R. Rackwitz (1983). First-order concepts in system reliability. *Struct. Safety* (1), 177–188.

Huang, M., C. Chan, & W. Lou (2012). Optimal performance-based design of wind sensitive tall buildings considering uncertainties. *Comput. Struct. 98–99*, 7–16.

Kiureghian, A. & T. Dakessian (1998). Multiple design points in 1st and 2nd-order reliability. *Str. Safety 20*(1), 37–49.

Lhomme, O. (1993). Consistency techniques for numeric CSPs. In *Proc. of the 13th IJCAI*, pp. 232–238.

Melchers, R. (1999). *Structural Reliability Analysis and Prediction*. Civil Engineering Series. John Wiley & Sons.

Moore, R. (1966). *Interval Analysis*. Prentice-Hall, Englewood Cliffs.

Nataf, A. (1962). Remarks on a multivariate transformation. *Comptes Rendus de l'Academie des Sciences 225*(1), 42–43.

Nikolaidis, E., D. Ghiocel, & S. Singhal (2007). *Engineering Design Reliability Applications: For the Aerospace, Automotive and Ship Industries*. CRC Press.

Rosenblatt, M. (1952). Remarks on a multivariate transformation. *Ann. Math. Stat. 3*(23), 470–472.

Sam-Haroud, D. & B. Faltings (1996). Consistency techniques for continuous constraints. *Constraints 1*(1/2): 85–118.

Sørensen, J. (2004). Notes in Structural Reliability Theory.

Wolfram Research, I. (2012). *Mathematica Edition: Version 9.0*. Champaign, Illinois: Wolfram Research, Inc.