ELSEVIER

Artificial Intelligence 96 (1997) 395–419

Artificial Intelligence

Tail uncertainty analysis in complex systems

Enrique Castillo *, Cristina Solares, Patricia Gómez

Department of Applied Mathematics and Computational Sciences, University of Cantabria, 39005 Santander, Spain

Received March 1996; revised May 1997

Abstract

The paper presents an efficient computational method for estimating the tails of a target variable Z which is related to other set of bounded variables $X = (X_1, \dots, X_n)$ by an increasing (decreasing) relation $Z = h(X_1, \dots, X_n)$. To this aim, variables X_i, $i = 1, \dots, n$ are sequentially simulated in such a manner that $Z = h(x_1, \dots, x_{i-1}, X_i, \dots, X_n)$ is guaranteed to be in the tail of Z. The method is shown to be very useful to perform an uncertainty analysis of Bayesian networks, when very large confidence intervals for the marginal/conditional probabilities are required, as in reliability or risk analysis. The method is shown to behave best when all scores coincide and is illustrated with several examples, including two examples of application to real cases. A comparison with the fast probability integration method, the best known method to date for solving this problem, shows that it gives better approximations. © 1997 Elsevier Science B.V.

Keywords: Bounded variables; Fast probability integration method; Likelihood weighing; Monotonic transformation; Tail simulation; Uncertainty analysis

1. Introduction

Consider the pressure tank in Fig. 1. It is a tank for storage of a pressurized fluid, which is introduced with the help of a pump activated by an electric motor. The tank is known not to have problems if the pump is working for periods of less than one minute. Therefore, a security mechanism, based on a time relay, F, interrupts the electric current after 60 seconds. In addition, a pressure switch, A, also interrupts the current if the pressure in the tank reaches a certain threshold value. The system also includes a switch, E, which initiates the operation of the system; one relay, D, which supplies the current after the initiation step and interrupts it after the activation of relay, F; and relay

* Corresponding author. Email: castie@ccaix3.unican.es.

PII S0004-3702(97)00052-0

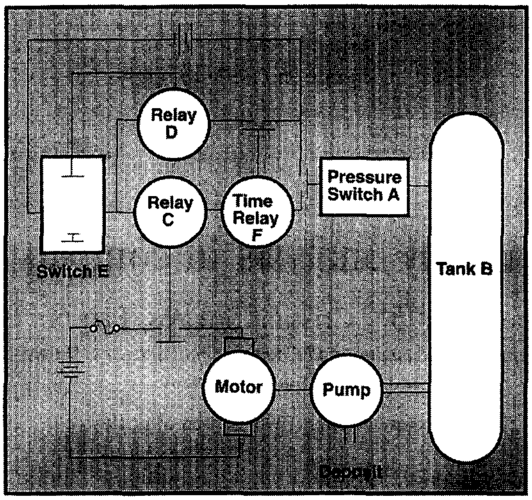

Fig. 1. Pressure tank diagram.

C, which starts the operation of the electrical circuit of the motor. Assume that we are interested in knowing the probability of failure of the pressure tank.

As shown in Section 7.2, using the techniques of fault tree analysis, the probability of failure of the tank can be approximated by

$$P(K=1) \approx x_1 + x_2 - x_1x_2 + x_3x_4 + x_3x_5 + x_3x_6, \tag{1}$$

where $x_1, \ldots, x_6$ are the probabilities of failure of its components.

When there are common causes of failure, fault trees become fault networks and then, Bayesian network models are more convenient to avoid replication of nodes. Bayesian networks are introduced in Section 2.

With common causes of failure, the probability of failure of the tank can be approximated by

$$P(K=1) \approx x_1 + 0.5x_2 - 0.5x_1x_2 + x_3x_4 + 2.5x_2x_5 + 0.25x_3x_6. \tag{2}$$

The preceding models (1) and (2) assume that the values of the probabilities $(x_1, x_2, \ldots)$ are known constant values. Thus, once these probabilities are known one can calculate the probability of failure using expressions (1) or (2).

However, in uncertainty analysis, these probabilities are assumed to be random variables (not known with certainty), and then the probability of failure becomes a random variable too. Thus, its associated density function is needed to obtain confidence intervals. In some cases, as in reliability or risk analysis of nuclear power plants, very high confidence intervals are required, which implies the estimation of their tails. In this paper we deal with a more general problem which consists of estimating the tail of a random variable which is related to other basic variables by a monotone relation. We assume that the cumulative distribution function (cdf) of the target variable is not directly available but determined through the basic variables. Note that in the case of Bayesian Networks the target variable is a marginal or conditional probability.

The usual procedure to solve the above problem is based on Monte Carlo simulation. Monte Carlo simulation is a very well known technique which allows dealing with random variables when they are related to other random variables by a complex relation. The common technique consists of simulating a large sample and using the empirical cumulative distribution function as an approximation to the exact one. The method performs well when we are interested in the central part of the distribution, but gives very poor approximations when the aim of the analysis is concentrated in the tails, as, for example, the estimation of small or large percentiles. However, in Engineering design only tails are important and one has to deal with extreme percentiles. In fact, the engineer is only interested in the occurrences of either very large values of magnitudes (temperatures, winds, waves, earthquakes, etc.) or very low values of the same magnitudes, because they produce structural, supply or environmental problems. This has motivated the appearance of extreme value theory in general (see [10] or [4]) and several papers dealing with the estimation of large percentiles in particular (see, for example, [14,19]).

Several simulation methods have been proposed for simulating random samples in Bayesian networks, as: stochastic simulation [13], likelihood weighing [17], hybrid methods of logic sampling and stochastic simulation [7], stratified sampling [2,6]. However, these methods have not been designed to solve the tail estimation problem.

We note here that estimation of extreme percentiles is difficult from real samples but if we can control the simulation method things are completely different. In this paper we present a method which allows simulating the tails of the target variable with a null proportion of rejections. This means that each data point in the simulated sample belongs to the desired tail.

A very good alternative to simulation is the fast probability integration (FPI) method. It appeared in the field of structural reliability with Freudenthal [9] in 1956, and has been expanded by Hasofer and Lind [12], Rackwitz and Flessler [15], and Breitung [3].

The main idea of the FPI method consists of approximating multidimensional integrals by known integrals, associated with standardized and independent multinomial random variables. Thus, eliminating the need to perform a numerical integration.

The FPI method has shown to give precise results and has demonstrated to be much more efficient than Monte Carlo simulation techniques for estimating extreme percentiles (see, for example, [11,18]).

Since the FPI is considered to be the best known method, to date, it will be used here for the sake of comparison with the proposed method.

The paper is structured as follows. In Section 2 we give some basic concepts which are necessary to understand the rest of the paper. In Section 3 we give a detailed description of the proposed method, which is illustrated in Section 4 by its application to several cases of monotone functions. Estimates are compared with exact values and those obtained with the FPI method. An improvement of this method, obtained by using simulation procedures with equal scores, is illustrated in Section 5. In Section 6 we adapt the method for its application to Bayesian networks, and in Section 7 we present two applications of the proposed method to real examples. Finally, Section 8 gives some conclusions.

2. Some required background

In this section we introduce some basic concepts to be used later.

2.1. Bayesian networks

Let $X = \{X_1, X_2, \ldots, X_n\}$ be a set of n discrete variables, taking values in the set $\{0, 1, \ldots, r_i\}$. A Bayesian network over X is a pair (D, P), where the graph D is a directed acyclic graph (DAG) over X and $P = \{P_1(x_1 \mid \pi_1), \ldots, P_n(x_n \mid \pi_n)\}$ is a set of n conditional probabilities, one for each variable, where Π_i is the set of parents of node X_i. The joint probability density of X can be written as:

$$P(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} P_i(x_i \mid \pi_i). \tag{3}$$

If we denote

$$\theta_{ij\pi} = P_i(X_i = j \mid \Pi_i = \pi), \quad j \in \{0, \ldots, r_i\}, \tag{4}$$

where π is any possible instantiation of the parents of X_i. The joint probability density (3) can be considered as a θ-parametric family.

Castillo, Gutiérrez and Hadi [5,6] have shown that the marginal $P(X_i = j)$, and conditional probabilities $P(X_i = j \mid E = e)$, where E is a set of evidential nodes with known values e, are polynomials and quotients of polynomials, respectively of the θ-parameters which are first degree in each parameter. This implies that they are monotone functions of the parameters.

2.2. A general simulation framework

Next, we illustrate a general simulation scheme which is sustained by the following theorem (see [16], [8] or [1]).

Theorem 1 (Rejection method). *Let X be a k-dimensional random vector with probability density $P_B(x)$. Suppose that $P_B(x)$ can be represented in the form*

$$P_B(x) = cg(x)P_S(x), \tag{5}$$

where $c > 1$, $0 < g(x) < 1$, and $P_S(x)$ is a probability density function (pdf). Let U be a standard uniform $U(0,1)$ random variable and let Y be a random variable with pdf $P_S(y)$ independent of U. It follows that the conditional distribution of Y given that $U \leqslant g(Y)$ coincides with the distribution of X.

This theorem leads to the *rejection method* which consists of the following steps:

- **Step 1**: Simulate Y using $P_S(y)$.
- **Step 2**: Draw a random uniform $U(0,1)$ number U.
- **Step 3**: If $U \leqslant g(Y)$ use Y as a random number from $P_B(x)$; otherwise, repeat the process from Step 1.

Table 1
Sampling distribution, and generation and scoring methods associated with some of the four most important simulation methods: (AR: Acceptance–Rejection, US: Uniform sampling, LW: Likelihood weighing, and MS: Markov sampling)

Method	Sampling distribution $P_S(x_i)$		Generation ordering	Score
	Evidential node	Non-evidential node		
AR	$P(X_i \mid \Pi_i)$	$P(X_i \mid \Pi_i)$	ancestral	1 or 0
US	$P_0(x_i)$	$1/\mathrm{card}(X_i)$	any	$\prod_{i=1}^{n} P(X_i \mid \Pi_i)$
LW	$P_0(x_i)$	$P_e(X_i \mid \Pi_i)$	ancestral	$\prod_{X_i \in E}^{n} P_e(e_i \mid \Pi_i)$
MS	$P_0(x_i)$	$P(X_i \mid \Pi_i) \prod_{X_j \in C_i}^{n} P_e(x_j \mid \Pi_j)$	any	1

Eq. (5) suggests the following modification of the rejection sampling method: Instead of rejecting a sample, we give it a probability proportional to $g(Y)$ or $cg(Y)$ (the score) and at the end we normalize to get a probability. This leads to a much higher efficiency of the simulation process, because no sample is rejected. The price we need to pay is that we have to calculate and store the scores.

Note that $P_S(x)$ is the sampling distribution and $cg(x) = P_B(x)/P_S(x)$ is the score.

Of a special interest is the case in which the joint probability density function (pdf) $f(x_1, \ldots, x_n)$ is known through a set $\{f(x_1), f(x_2 \mid x_1), \ldots, f(x_n \mid x_1, \ldots, x_{n-1})\}$ of conditional probabilities, as in the case of Bayesian networks.

In known simulation methods for Bayesian networks, the sampling distribution can be written as the product of the sampling distributions of the nodes. So, let $P_S(x_i)$ be the sampling distribution of node x_i, then

$$P_S(X) = \prod_{i=1}^{n} P_S(x_i). \tag{6}$$

The real joint probability can be written as

$$P_B(X) = \prod_{i=1}^{n} P_B(x_i \mid \pi_i). \tag{7}$$

If some of these factors coincide, they cancel in the score expression

$$w = \frac{P_B(x)}{P_S(x)} = \prod_{i=1}^{n} \frac{P_B(x_i \mid \pi_i)}{P_S(x_i)}. \tag{8}$$

Sampling algorithms for Bayesian networks consist of the following three components:

(1) A sampling distribution,
(2) An instantiation generation method, and
(3) A scoring method.

All the existing methods show differences in one of these three elements. Table 1 shows the sampling distribution, the generation method and the scoring methods associated with four of the most important simulation methods (see [6, Chapter 9], or the references given in the Introduction), where

$$P_0(x_i) = \begin{cases} 1 & \text{if } x_i = e_i, \\ 0 & \text{otherwise,} \end{cases}$$

C_i is the set of sons of X_i, $P_e()$ means $P()$ with all evidential nodes instantiated to their corresponding evidential values e_i, card(X_i) is the cardinal (number of elements) of X_i and E is the set of evidential nodes.

3. The proposed method

The main idea of the proposed methods consists of simulating only the tail of the target variable. Assume that $Z = h(\boldsymbol{X})$, that is, we have a bounded random variable Z related to a set $\boldsymbol{X} = \{X_1, \dots, X_n\}$ of basic random variables by an increasing (decreasing) relation.

Note that if $h(\boldsymbol{X})$ is decreasing, we can work with $-Z$ instead of Z. Assume also that $\infty \leqslant a_i \leqslant X_i \leqslant b_i \leqslant \infty$. Then, $h(\boldsymbol{a}) \leqslant Z \leqslant h(\boldsymbol{b})$, that is, $h(\boldsymbol{a})$ and $h(\boldsymbol{b})$ are the lower and upper bounds of Z, respectively.

We start by analyzing the right tail and then we give the required modifications for the left tail.

3.1. Right tail

Our aim is to approximate the cdf in the tail $\mathcal{T} = \{z \mid h(\boldsymbol{b}) - \varepsilon < z \leqslant h(\boldsymbol{b})\}$ of Z, by simulating the random variable $(X_1, \dots, X_n)$ restricted to $Z > h(\boldsymbol{b}) - \varepsilon$.

For any i, let us denote $h_i^{-1}(\boldsymbol{x}_{i-1}, z, \boldsymbol{x}^{i+1})$ the inverse of $h(\boldsymbol{x})$ with respect to x_i, where we have denoted $\boldsymbol{x}_i = (x_1, \dots, x_i)$ and $\boldsymbol{x}^i = (x_i, \dots, x_n)$.

The proposed method sequentially simulates variables $X_1, \dots, X_n$ in the following form. Assume that we have already simulated variables $X_1 = x_1, \dots, X_{i-1} = x_{i-1}$. Then we simulate X_i such that

$$h(\boldsymbol{b}) - \varepsilon < h(\boldsymbol{x}_{i-1}, X_i, \boldsymbol{X}^{i+1}) \leqslant h(\boldsymbol{x}_{i-1}, \boldsymbol{b}^i). \tag{9}$$

Note that once variables $\boldsymbol{X}_{i-1}$ have been simulated, the new upper bound of Z is $h(\boldsymbol{x}_{i-1}, \boldsymbol{b}^i)$.

From (9) we get

$$l_i(\boldsymbol{X}^{i+1}) < X_i \leqslant u_i(\boldsymbol{X}^{i+1}), \tag{10}$$

where

$$l_i(\boldsymbol{X}^{i+1}) = h_i^{-1}(\boldsymbol{x}_{i-1}, h(\boldsymbol{b}) - \varepsilon, \boldsymbol{X}^{i+1}) \tag{11}$$

and

$$u_i(\boldsymbol{X}^{i+1}) = h_i^{-1}(\boldsymbol{x}_{i-1}, h(\boldsymbol{x}_{i-1}, \boldsymbol{b}^i), \boldsymbol{X}^{i+1}) \tag{12}$$

Since $\boldsymbol{X}^{i+1}$ have not been simulated yet, we must choose the largest possible interval which, taking into account the constraint $a_i < X_i \leqslant b_i$, is

$$L_i < X_i \leqslant U_i, \tag{13}$$

where

$$L_i = \max\left[\min_{\boldsymbol{X}^{i+1}} l_i(\boldsymbol{X}^{i+1}), a_i\right] = \max(h_i^{-1}(\boldsymbol{x}_{i-1}, h(\boldsymbol{b}) - \varepsilon, \boldsymbol{b}^{i+1}), a_i) \tag{14}$$

and

$$U_i = \min\left[\max_{\boldsymbol{X}^{i+1}} u_i(\boldsymbol{X}^{i+1}), b_i\right] = \min(h_i^{-1}(\boldsymbol{x}_{i-1}, h(\boldsymbol{x}_{i-1}, \boldsymbol{b}^i), \boldsymbol{a}^{i+1}), b_i) \tag{15}$$

are the lower and upper bounds of X_i given $\boldsymbol{X}_i$.

Once L_i and U_i are known, we can sequentially simulate X_i, $i = 1, \ldots, n$, with density proportional to $P_S(x_i; \boldsymbol{x}_{i-1})$ in the region $L_i < X_i \leqslant U_i$. Note that different choices of $P_S(x_i; \boldsymbol{x}_{i-1})$ lead to different methods.

3.2. Left tail

Similarly, we can approximate the cdf in the left tail $\mathcal{T} = \{z \mid h(\boldsymbol{a}) < z \leqslant h(\boldsymbol{a}) + \varepsilon\}$ of Z, by simulating the random variable $\boldsymbol{X}$ restricted to $Z \leqslant h(\boldsymbol{a}) + \varepsilon$.

The above discussion, expressions and method for the right tail remain valid for the left tail with the following changes:

- Expression (9) becomes

$$h(\boldsymbol{x}_{i-1}, \boldsymbol{a}^i) < h(\boldsymbol{x}_{i-1}, X_i, \boldsymbol{X}^{i+1}) \leqslant h(\boldsymbol{a}) + \varepsilon, \tag{16}$$

because, once variables $\boldsymbol{X}_{i-1}$ have been simulated, the new lower bound of Z is $h(\boldsymbol{x}_{i-1}, \boldsymbol{a}^i)$.

- Expression (11) becomes

$$l_i(\boldsymbol{X}^{i+1}) = h_i^{-1}(\boldsymbol{x}_{i-1}, h(\boldsymbol{x}_{i-1}, \boldsymbol{a}^i), \boldsymbol{X}^{i+1}). \tag{17}$$

- Expression (12) becomes

$$u_i(\boldsymbol{X}^{i+1}) = h_i^{-1}(\boldsymbol{x}_{i-1}, h(\boldsymbol{a}) + \varepsilon, \boldsymbol{X}^{i+1}). \tag{18}$$

- Expression (14) becomes

$$L_i = \max\left[\min_{\boldsymbol{X}^{i+1}} l_i(\boldsymbol{X}^{i+1}), a_i\right] = \max(h_i^{-1}(\boldsymbol{x}_{i-1}, h(\boldsymbol{a}), \boldsymbol{b}^{i+1}), a_i). \tag{19}$$

- Expression (15) becomes

$$U_i = \min\left[\max_{\boldsymbol{X}^{i+1}} u_i(\boldsymbol{X}^{i+1}), b_i\right] = \min(h_i^{-1}(\boldsymbol{x}_{i-1}, h(\boldsymbol{a}) + \varepsilon, \boldsymbol{a}^{i+1}), b_i). \tag{20}$$

3.2.1. Simulation algorithm

The proposed method can be summarized in the following algorithm:

Algorithm 1.

Input:

- An increasing function defining the target variable: $Z = h(\boldsymbol{X})$
- A set of n conditional probabilities $P_{\mathrm{B}}(X_i \mid \boldsymbol{X}_{i-1})$.
- Lower and upper bounds of the basic random variables $\boldsymbol{X}$: $\boldsymbol{a}$ and $\boldsymbol{b}$.
- Sample size m and desired departure ε from lower $h(\boldsymbol{a})$ or upper bound $h(\boldsymbol{b})$ of the target random variable Z.

Output:

- A sample of size m from the left tail $\mathcal{T} = \{z \mid h(\boldsymbol{a}) < z \leqslant h(\boldsymbol{a}) + \varepsilon\}$ or the right tail $\mathcal{T} = \{z \mid h(\boldsymbol{b}) - \varepsilon < z \leqslant h(\boldsymbol{b})\}$ of the target Z.

Steps:

- **Step 1**: Simulate sequentially X_i, $i = 1, \ldots, n$ in the interval

$$L_i < X_i \leqslant U_i, \tag{21}$$

  using $P_{\mathrm{S}}(x_i; \boldsymbol{x}_{i-1})$, that is, we simulate truncated variables, where x_i, $i = 1, \ldots, n$ are the simulated values.
- **Step 2**: Calculate the simulated sample value $z_j = h(\boldsymbol{x})$ and assign it the score

$$w_j = \prod_{i=1}^{n} \frac{P_{\mathrm{B}}(x_i \mid \boldsymbol{x}_{i-1})}{P_{\mathrm{S}}(x_i; \boldsymbol{x}_{i-1})}. \tag{22}$$

- **Step 3**: Store the pair (z_j, w_j).
- **Step 4**: Repeat steps 1 to 3 m times.
- **Step 5**: Sort the pairs (z_j, w_j), $j = 1, \ldots, m$ with respect to z_j.
- **Step 6**: Replace w_j in the pairs (z_j, w_j), $j = 1, \ldots, m$ by $\frac{1}{m}\sum_{k=1}^{j} w_k$, for the left tail and by $1 - \frac{1}{m}\sum_{k=j+1}^{m} w_k$, for the right tail.
- **Step 7**: The resulting w_j, $j = 1, \ldots, m$ are the simulated approximations $\hat{F}_Z(z_j)$ to $F_Z(z_j)$.

Note that many choices are possible for $P_{\mathrm{S}}(x_i; \boldsymbol{x}_{i-1})$ in Step 1. Two special cases are:

(1) *Likelihood weighing method*: In this case the sampling distribution is

$$P_{\mathrm{S}}(x_i) = \frac{P_{\mathrm{B}}(x_i \mid \boldsymbol{x}_{i-1})}{F_{X_i|\boldsymbol{X}_{i-1}}(U_i \mid \boldsymbol{x}_{i-1}) - F_{(X_i|\boldsymbol{X}_{i-1})}(L_i \mid \boldsymbol{x}_{i-1})} \tag{23}$$

and then the score (8) becomes:

$$w = \frac{P_{\mathrm{B}}(x)}{P_{\mathrm{S}}(x)} = \frac{\prod_{i=1}^{n} P_{\mathrm{B}}(x_i \mid \pi_i)}{\prod_{i=1}^{n} P_{\mathrm{S}}(x_i \mid \pi_i)}$$
$$= \prod_{i=1}^{n} [F_{X_i|\boldsymbol{X}_{i-1}}(U_i \mid \boldsymbol{x}_{i-1}) - F_{X_i|\boldsymbol{X}_{i-1}}(L_i \mid \boldsymbol{x}_{i-1})]. \tag{24}$$

(2) *Uniform sampling method*: In this case the sampling distribution is

$$P_S(x_i) = \frac{1}{U_i - L_i} \tag{25}$$

and then the score (8) becomes:

$$w = \frac{P_B(x)}{P_S(x)} = \frac{\prod_{i=1}^n P_B(x_i \mid \pi_i)}{\prod_{i=1}^n P_S(x_i \mid \pi_i)} = \prod_{i=1}^n [(U_i - L_i)\exp(x_i)]. \tag{26}$$

4. Experimental design and performance of the method

In this section we study the performance of the proposed method by its application to several examples and the corresponding analysis. To this end, the experimental design has been based on the following considerations:

(1) Several examples with known associated cdf for Z have been selected so that a comparison with the exact cdf $F_Z(z)$ of the target variable Z be possible.

(2) A set of simulation runs have been performed. In each experiment, a sample of size m has been simulated and the corresponding $\hat{F}_Z(z)$ has been calculated.

(3) To measure the quality of fit the following error statistics have been used:

$$Q_1 = \max_i \mid F(z_i) - \hat{F}(z_i) \mid, \tag{27}$$

and

$$Q_2 = \left\{\frac{1}{m}\sum_{i=1}^{m}(F(z_i) - \hat{F}(z_i))^2\right\}^{1/2}, \tag{28}$$

where $F(z)$ and $\hat{F}(z)$ are the exact and the estimated cdf of Z, respectively.

(4) To estimate the mean and standard deviation of Q_1 and Q_2, the experiments have been replicated 1000 times. The corresponding estimates appear in the tables.

(5) Sample sizes $m = 100$, 200, 500 and 1000 have been selected.

(6) The results have been compared with the corresponding FPI values. To this end, the mean error values associated with this method and the corresponding intervals are also shown in all tables. Note that these estimates do not depend on the sample size.

(7) No comparison with the standard Monte Carlo method has been done, because it is very inefficient for simulating the tails. In fact, a very large proportion of sample values are out of the tail when this involves high percentiles.

In the following subsections some of the conducted experiments are described.

4.1. Right tail of sums of uniforms

In the case of sums $h(X) = \sum_{i=1}^n X_i$, and the functions $l_i(X^{i+1})$ and $u_i(X^{i+1})$ in (11) and (12)become

Table 2
Right tail of the sum of four uniforms: Mean and standard deviation estimates of Q_1 and Q_2 and the corresponding values of the FPI method, for samples sizes of 100, 200, 500 and 1000 corresponding to the variable interval $(3.1, 4)$ or the cdf interval $(0.9727, 1)$ (1000 replications)

m	Proposed method				FPI method	
	$\hat{E}[Q_1]$	$\hat{\sigma}[Q_1]$	$\hat{E}[Q_2]$	$\hat{\sigma}[Q_2]$	$E[Q_1]$	$E[Q_2]$
100	0.00703	0.00316	0.00462	0.00298	0.0400	0.0166
200	0.00518	0.00214	0.00334	0.00207	0.0400	0.0166
500	0.00329	0.00139	0.00208	0.00137	0.0400	0.0166
1000	0.00237	0.00091	0.00147	0.00091	0.0400	0.0166

$$\begin{aligned} l_i(X^{i+1}) &= -\sum_{k=1}^{i-1} x_k + \sum_{k=1}^{n} b_k - \varepsilon - \sum_{k=i+1}^{n} X_k, \\ u_i(X^{i+1}) &= \sum_{k=i}^{n} b_k - \sum_{k=i+1}^{n} X_k. \end{aligned} \tag{29}$$

Thus, from (14) and (15) we get

$$L_i = \max\left(\sum_{k=1}^{i} b_k - \sum_{k=1}^{i-1} x_k - \varepsilon, a_i\right), \qquad U_i = b_i. \tag{30}$$

The exact cumulative distribution function of the sum of n uniforms is given by

$$F(z) = \frac{1}{n!}\sum_{r=0}^{[z]}(-1)^r\binom{n}{r}(z-r)^n, \quad 0 < z < n, \tag{31}$$

where $[z]$ is the integer part of z.

Table 2 shows the mean and standard deviation estimates of Q_1 and Q_2 and the corresponding values of the FPI method, for $n = 4$ and samples sizes m of 100, 200, 500 and 1000, obtained by the likelihood weighing method.

4.2. Right tail of products of uniforms

In the case of products $h(X) = \prod_{i=1}^{n} X_i$. If we assume $X_i \geqslant 0$, $i = 1, \ldots, n$, the functions $l_i(X^{i+1})$ and $u_i(X^{i+1})$ in (11) and (12) become

$$l_i(X^{i+1}) = \frac{\prod_{k=1}^{n} b_k - \varepsilon}{(\prod_{k=1}^{i-1} x_k)(\prod_{k=i+1}^{n} X_k)}, \tag{32}$$

$$u_i(X^{i+1}) = \frac{\prod_{k=i}^{n} b_k}{\prod_{k=i+1}^{n} X_k}. \tag{33}$$

Thus, L_i and U_i are (see (14) and (15))

Table 3
Right tail of the product of three uniforms: Mean and standard deviation estimates of Q_1 and Q_2 and the corresponding values of the FPI method, for samples sizes of 100, 200, 500 and 1000 corresponding to the variable interval $(0.8, 1)$ or the cdf interval $(0.9984, 1)$ (1000 replications)

m	Proposed method				FPI method	
	$\hat{E}[Q_1]$	$\hat{\sigma}[Q_1]$	$\hat{E}[Q_2]$	$\hat{\sigma}[Q_2]$	$E[Q_1]$	$E[Q_2]$
100	0.000239	0.000095	0.000141	0.000081	0.00430	0.00171
200	0.000178	0.000067	0.000104	0.000057	0.00430	0.00171
500	0.000113	0.000041	0.000064	0.000036	0.00430	0.00171
1000	0.000080	0.000030	0.000046	0.000026	0.00430	0.00171

$$L_i = \max\left(\frac{\prod_{k=i}^{n} b_k - \varepsilon}{(\prod_{k=1}^{i-1} x_k)(\prod_{k=i+1}^{n} b_k)}, a_i\right), \tag{34}$$

$$U_i = b_i. \tag{35}$$

The exact cumulative distribution function of the product of n uniforms $U(0,1)$ is

$$F(z) = z\left(\sum_{i=0}^{n-1} \frac{(-\log z)^i}{i!}\right), \quad 0 \leqslant z \leqslant 1. \tag{36}$$

Table 3 shows the mean and standard deviation estimates of Q_1 and Q_2 and the corresponding values of the FPI method, for $n = 3$ and samples sizes m of 100, 200, 500 and 1000, obtained by the likelihood weighing method.

4.3. Right tail of sums of products of uniforms

As a simple example we consider here the case

$$h(X_1, \ldots, X_6) = X_1 + X_2 + X_3(X_4 + X_5 + X_6). \tag{37}$$

and we assume that $X_i \sim U(0, \rho)$, $i = 1, \ldots, 6$ and with $\rho < \frac{1}{3}$ and $\varepsilon < 2\rho + 3\rho^2$, which implies $b_i = \rho$, $i = 1, \ldots, 6$.

The $l_i(X^{i+1})$ and $u_i(X^{i+1})$ functions become the functions in Table 4 and L_i and U_i become the values in Table 5.

Fig. 2 shows the exact and simulated tail (0.003,004) of the random variable $Z = X_1 + X_2 + X_3(X_4 + X_5 + X_6)$, when we assume that X_i, $i = 1, \ldots, n$ are independent and identically distributed (i.i.d.) uniforms $U(0, 0.002)$; $\varepsilon = 0.001$, which implies $\rho = 0.002$.

4.4. Left tail of products

In the case of products of non-negative variables $h(X) = \prod_{i=1}^{n} X_i$, and the functions $l_i(X^{i+1})$ and $u_i(X^{i+1})$ in (17) and (18) become

$$l_i(X^{i+1}) = \frac{\prod_{k=i}^{n} a_k}{(\prod_{k=i+1}^{n} X_k)}. \tag{38}$$

Table 4
Lower $l_i(X^{i+1})$ and upper $u_i(X^{i+1})$ bounds for the sequential simulation of sums of products

i	$l_i(X^{i+1})$	$u_i(X^{i+1})$
1	$2\rho + 3\rho^2 - \varepsilon - X_2 - X_3(X_4 + X_5 + X_6)$	$2\rho + 3\rho^2 - X_2 - X_3(X_4 + X_5 + X_6)$
2	$2\rho + 3\rho^2 - \varepsilon - x_1 - X_3(X_4 + X_5 + X_6)$	$\rho + 3\rho^2 - X_3(X_4 + X_5 + X_6)$
3	$\frac{2\rho+3\rho^2-\varepsilon-x_1-x_2}{X_4+X_5+X_6}$	$\frac{3\rho^2}{X_4+X_5+X_6}$
4	$\frac{2\rho+3\rho^2-\varepsilon-x_1-x_2}{x_3} - X_5 - X_6$	$3\rho - X_5 - X_6$
5	$\frac{2\rho+3\rho^2-\varepsilon-x_1-x_2}{x_3} - x_4 - X_6$	$2\rho - X_6$
6	$\frac{2\rho+3\rho^2-\varepsilon-x_1-x_2}{x_3} - x_4 - x_5$	ρ

Table 5
Lower L_i and upper U_i bounds for the sequential simulation of sums of products

i	L_i	U_i
1	$\max(\rho - \varepsilon, 0)$	ρ
2	$\max(2\rho - \varepsilon - x_1, 0)$	ρ
3	$\max(\frac{2\rho+3\rho^2-\varepsilon-x_1-x_2}{3\rho}, 0)$	ρ
4	$\max(\frac{2\rho+3\rho^2-\varepsilon-x_1-x_2}{x_3} - 2\rho, 0)$	ρ
5	$\max(\frac{2\rho+3\rho^2-\varepsilon-x_1-x_2}{x_3} - x_4 - \rho, 0)$	ρ
6	$\max(\frac{2\rho+3\rho^2-\varepsilon-x_1-x_2}{x_3} - x_4 - x_5, 0)$	ρ

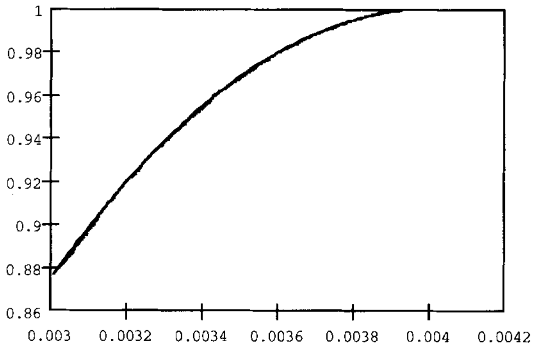

Fig. 2. Exact and simulated tail (0.003,004) of $Z = X_1 + X_2 + X_3(X_4 + X_5 + X_6)$.

$$u_i(X^{i+1}) = \frac{\prod_{k=1}^{n} a_k + \varepsilon}{\prod_{k=1}^{i-1} x_k \prod_{k=i+1}^{n} X_k}. \tag{39}$$

Thus, L_i and U_i become (see (14) and (15))

$$L_i = \max\left(\min_{X^{i+1}} \frac{\prod_{k=i}^{n} a_k}{\prod_{k=i+1}^{n} X_k}, a_i\right) = a_i, \tag{40}$$

Table 6
Left tail of the sum of four exponentials: Mean and standard deviation estimates of Q_1 and Q_2 and the corresponding values of the FPI method, for samples sizes of 100, 200, 500 and 1000 corresponding to the variable interval (0, 0.9) or the cdf interval (0, 0.0135) (1000 replications) (Likelihood weighing and uniform sampling methods)

Likelihood weighing						
m	Proposed method				FPI method	
	$\hat{E}[Q_1]$	$\hat{\sigma}[Q_1]$	$\hat{E}[Q_2]$	$\hat{\sigma}[Q_2]$	$E[Q_1]$	$E[Q_2]$
100	0.00257	0.00106	0.00159	0.00095	0.0376	0.0153
200	0.00186	0.00076	0.00113	0.00070	0.0376	0.0153
500	0.00122	0.00050	0.00074	0.00046	0.0376	0.0153
1000	0.00086	0.00035	0.00051	0.00032	0.0376	0.0153
Uniform sampling						
m	Proposed method				FPI method	
	$\hat{E}[Q_1]$	$\hat{\sigma}[Q_1]$	$\hat{E}[Q_2]$	$\hat{\sigma}[Q_2]$	$E[Q_1]$	$E[Q_2]$
100	0.00380	0.00171	0.00255	0.00166	0.0376	0.0153
200	0.00275	0.00118	0.00179	0.00116	0.0376	0.0153
500	0.00182	0.00076	0.00116	0.00076	0.0376	0.0153
1000	0.00130	0.00056	0.00083	0.00056	0.0376	0.0153

$$U_i = \min\left(\frac{\prod_{k=1}^{n} a_k + \varepsilon}{\prod_{k=1}^{i-1} x_k \prod_{k=i+1}^{n} a_k}, b_i\right). \tag{41}$$

4.5. Left tail of sums of exponentials

In the case of sum of four exponential $E(1)$ variables, Table 6 shows the mean and standard deviation estimates of Q_1 and Q_2 and the corresponding values of the FPI method, for samples sizes m of 100, 200, 500 and 1000 when the likelihood weighing and the uniform sampling methods are used. Note that the first gives better results.

4.6. Analysis and discussion of experimental results

From the preceding analysis, we can conclude the following:

(1) Percentiles corresponding to very low or high probabilities are very closely estimated with a sample of relatively small size using the proposed method.
(2) The quality of the approximation improves substantially with the sample size m.
(3) The method seems to work well for any percentile (see previous examples).
(4) In all conducted experiments, the proposed method gave much better results than the FPI method.
(5) The likelihood weighing method seems to give better results than the uniform sampling.
(6) The proposed method deteriorates for increasing number of basic variables.

5. Improved methods

The no rejection method performs very well for small number of basic variables but deteriorates if this number increases. The main reason for this is that the number of feasible different instantiations blows up with the number of basic variables and the associated scores become very far apart. This problem can be solved if we design simulating procedures leading to similar scores for all instantiations, the ideal situation being all scores to be equal. Now we show how this optimal solution can be theoretically achieved.

If we choose

$$\begin{aligned} p_{\mathrm{S}}(x_i) &= \frac{p_{\mathrm{B}}(x_i \mid \boldsymbol{x}_{i-1})\,[G_{i+1}(U_{i+1}) - G_{i+1}(L_{i+1})]}{G_i(U_i) - G_i(L_i)}, \\ p_{\mathrm{S}}(x_n) &= \frac{p_{\mathrm{B}}(x_n \mid \boldsymbol{x}_{n-1})}{G_n(U_n) - G_n(L_n)}, \end{aligned} \tag{42}$$

where $G_i(x_i)$ is the cdf associated with the pdf

$$\begin{aligned} &g_i(x) \doteq p_{\mathrm{B}}(x \mid \boldsymbol{x}_{i-1})\,[G_{i+1}(U_{i+1}) - G_{i+1}(L_{i+1})], \\ &\quad L_i \leqslant x \leqslant U_i, \;\; i = 1, \ldots, n-1, \\ &g_n(x) = p_{\mathrm{B}}(x \mid \boldsymbol{x}_{n-1}), \quad L_n \leqslant x \leqslant U_n, \end{aligned} \tag{43}$$

the score becomes $w = G_1(U_1) - G_1(L_1)$, which is independent of the sample and implies constant scores.

The method is illustrated with the two examples below.

5.1. Sum of uniforms (left tail)

Assume that we consider the random variable $Z = \sum_{i=1}^{n} X_i$, where X_i, $i = 1, \ldots, n$ are i.i.d uniform $U(0,1)$ random variables. Assume also that we want to simulate the left tail of Z. In this case we can simulate with (42) to obtain:

$$L_i = a_i = 0, \tag{44}$$

$$U_i = \min\left(\varepsilon - \sum_{k=1}^{i-1} x_i, b_i\right), \tag{45}$$

$$G_n(x) = \varepsilon - x_1 - \cdots - x_{n-1} - x; \; L_n \leqslant x \leqslant U_n, \tag{46}$$

$$G_i(x) = 1 - \left(1 - \frac{x}{\varepsilon - \sum_{k=1}^{i-1} x_k}\right)^{n-i+1} \tag{47}$$

and the score becomes

$$w = G_1(U_1) - G_1(L_1) = G_1(\varepsilon) - G_1(0) = \varepsilon^n/n!. \tag{48}$$

Thus, we simulate using the expressions:

Table 7
Right tail of the sum of four uniforms: Mean and standard deviation estimates of Q_1 and Q_2 and the corresponding values of the FPI method, for samples sizes of 100, 200, 500 and 1000 corresponding to the variable interval $(3.1,4)$ or the cdf interval $(0.9727,1)$ (1000 replications) (likelihood weighing and improved methods)

	Likelihood weighing					
m	Proposed method				FPI method	
	$\hat{E}[Q_1]$	$\hat{\sigma}[Q_1]$	$\hat{E}[Q_2]$	$\hat{\sigma}[Q_2]$	$E[Q_1]$	$E[Q_2]$
100	0.00703	0.00316	0.00462	0.00298	0.0400	0.0166
200	0.00518	0.00214	0.00334	0.00207	0.0400	0.0166
500	0.00329	0.00139	0.00208	0.00137	0.0400	0.0166
1000	0.00237	0.00091	0.00147	0.00091	0.0400	0.0166
	Uniform sampling					
m	Proposed method				FPI method	
	$\hat{E}[Q_1]$	$\hat{\sigma}[Q_1]$	$\hat{E}[Q_2]$	$\hat{\sigma}[Q_2]$	$E[Q_1]$	$E[Q_2]$
100	0.00220	0.00070	0.00103	0.00041	0.0400	0.0166
200	0.00156	0.00049	0.00071	0.00029	0.0400	0.0166
500	0.00102	0.00032	0.00046	0.00019	0.0400	0.0166
1000	0.00072	0.00022	0.00032	0.00013	0.0400	0.0166

$$X_i = \Big(\varepsilon - \sum_{k=1}^{i-1} x_k\Big)\big(1 - p_i^{1/(n-i+1)}\big), \quad i = 1,\ldots,n, \tag{49}$$

where p_i are i.i.d. $U(0,1)$ random variables and we have assumed $\varepsilon < 1$.

5.2. Sum of uniforms (right tail)

Assume that we consider the random variable $Z = \sum_{i=1}^{n} X_i$, where X_i, $i = 1,\ldots,n$ are i.i.d uniform $U(0,1)$ random variables. Assume also that we want to simulate the right tail of Z. In this case the score becomes

$$w = \varepsilon^n/n! \tag{50}$$

and we simulate with the expressions:

$$X_i = \Big(i - \varepsilon - \sum_{k=1}^{i-1} x_k\Big) + \Big[1 - \Big(i - \varepsilon - \sum_{k=1}^{i-1} x_k\Big)\Big] p_i^{1/(n-i+1)}, \quad i = 1,\ldots,n, \tag{51}$$

where p_i are i.i.d. $U(0,1)$ random variables and we have assumed $\varepsilon < 1$.

Table 7 shows the mean and standard deviation estimates of Q_1 and Q_2 and the corresponding values of the FPI method, for samples sizes m of 100, 200, 500 and 1000, together with a duplicate of the results in Table 2. In both cases the simulation has been done by the likelihood weighing method. A comparison of both methods shows that the improved method gives better results.

6. Adapting the method to uncertainty analysis in Bayesian networks

The main problem of the proposed method consists of finding the n inverse functions $h_i^{-1}(\boldsymbol{x}_{i-1}, z, \boldsymbol{x}^{i+1})$, $i = 1, \ldots, n$, since the functions can be extremely complex.

Fortunately, the algebraic expressions associated with uncertainty analysis in Bayesian Networks analysis consists of polynomials or quotients of polynomials of the basic variables involved, which are first degree in each of them. This allows avoiding the inversion of the target function for each variable. The idea is as follows.

In the case of polynomial, the algebraic expression for the unavailability $h(\boldsymbol{x}_n)$ can be written as

$$h(\boldsymbol{x}_{i-1}, x_i, \boldsymbol{x}^{i+1}) = \alpha(\boldsymbol{x}_{i-1}, \boldsymbol{x}^{i+1}) + x_i\beta(\boldsymbol{x}_{i-1}, \boldsymbol{x}^{i+1}), \tag{52}$$

where $\alpha(\boldsymbol{x}_{i-1}, \boldsymbol{x}^{i+1})$ and $\beta(\boldsymbol{x}_{i-1}, \boldsymbol{x}^{i+1})$ are the coefficients of the first degree polynomial function in x_i.

This linear function in x_i can be evaluated in two different points of x_i, a_i and b_i, say, to get

$$h(\boldsymbol{x}_{i-1}, a_i, \boldsymbol{x}^{i+1}) = \alpha(\boldsymbol{x}_{i-1}, \boldsymbol{x}^{i+1}) + a_i\beta(\boldsymbol{x}_{i-1}, \boldsymbol{x}^{i+1}), \tag{53}$$

$$h(\boldsymbol{x}_{i-1}, b_i, \boldsymbol{x}^{i+1}) = \alpha(\boldsymbol{x}_{i-1}, \boldsymbol{x}^{i+1}) + b_i\beta(\boldsymbol{x}_{i-1}, \boldsymbol{x}^{i+1}), \tag{54}$$

from which we can write

$$h(\boldsymbol{x}_{i-1}, x_i, \boldsymbol{x}^{i+1}) = \frac{h(\boldsymbol{x}_{i-1}, b_i, \boldsymbol{x}^{i+1})(x_i - a_i) + h(\boldsymbol{x}_{i-1}, a_i, \boldsymbol{x}^{i+1})(b_i - x_i)}{b_i - a_i}. \tag{55}$$

Thus, its inverse can be written as

$$h_i^{-1}(\boldsymbol{x}_{i-1}, u, \boldsymbol{x}^{i+1}) = \frac{b_i(u - h(\boldsymbol{x}_{i-1}, a_i, \boldsymbol{x}^{i+1})) + a_i(h(\boldsymbol{x}_{i-1}, b_i, \boldsymbol{x}^{i+1}) - u)}{h(\boldsymbol{x}_{i-1}, b_i, \boldsymbol{x}^{i+1}) - h(\boldsymbol{x}_{i-1}, a_i, \boldsymbol{x}^{i+1})}. \tag{56}$$

This means that we can write the inverse $h_i^{-1}(.)$ as a function of $h(.)$. In other words, we do not need to calculate the inverse.

Similarly, if we deal with rational functions, we have

$$h(\boldsymbol{x}_{i-1}, x_i, \boldsymbol{x}^{i+1}) = \frac{\alpha(\boldsymbol{x}_{i-1}, \boldsymbol{x}^{i+1}) + x_i\beta(\boldsymbol{x}_{i-1}, \boldsymbol{x}^{i+1})}{\gamma(\boldsymbol{x}_{i-1}, \boldsymbol{x}^{i+1}) + x_i\delta(\boldsymbol{x}_{i-1}, \boldsymbol{x}^{i+1})}. \tag{57}$$

If β or δ are not null we can divide numerator and denominator by it and we realize that (57) depends on three coefficients. Thus, we need to evaluate (57) in three different points to obtain an expression for the inverse, similar to (56).

Note that this has many computational advantages since we need to implement only functions $h(.)$ instead of all possible inverses.

7. Examples

In this section we show two examples of applications to probability risk assessment. This is one of the fields where the presented method fits very well, since tails play the most important role.

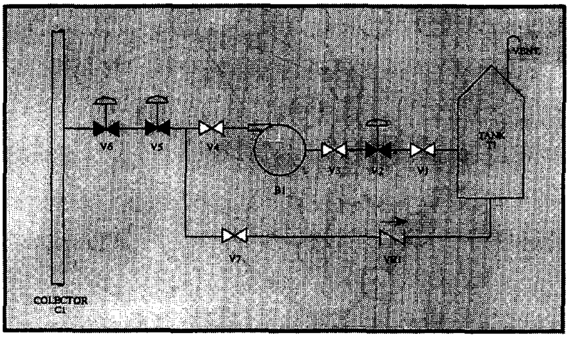

Fig. 3. Simplified standby system diagram.

Table 8
Variables associated with different element types of failures

Variable	Physical meaning	Variable	Physical meaning
ACA	Electric power failure	*G*15	Valve *V*1 does not receive water flow
*B*1*A*	Pump *B*1 fails to start	*M*1	Element out of service (maintenance)
*B*1*B*	Pump *B*1 fails after starting	*O*1	Valve *V*4 opened after maintenance
*G*1	Collector does not receive water flow	*O*2	Valve *V*3 opened after maintenance
*G*2	Valve *V*6 fails to open	*SISA*	Logic signal failure
*G*3	Valve *V*6 does not receive water flow	*T*1*A*	Tank failure
*G*4	Valve *V*5 fails to open	*T*1*B*	Ventilation tank failure
*G*5	Valve *V*5 does not receive water flow	*V*1*A*	Valve *V*1 is blocked
*G*6	Valve *V*4 is closed	*V*2*A*	Valve *V*2 mechanical failure
*G*7	Valve *V*4 does not receive water flow	*V*2*B*	Valve *V*2 is blocked
*G*8	Pump *B*1 failure	*V*3*A*	Valve *V*3 is blocked
*G*9	Pump *B*1 does not received water flow	*V*4*A*	Valve *V*4 is blocked
*G*10	Valve *V*3 is closed	*V*5*A*	Valve *V*5 mechanical failure
*G*11	Valve *V*3 does not receive water flow	*V*5*B*	Valve *V*5 is blocked
*G*12	Valve *V*2 fails to open	*V*6*A*	Valve *V*6 mechanical failure
*G*13	Valve *V*2 does not receive water flow	*V*6*B*	Valve *V*6 is blocked

7.1. Example 1: Standby system

The simplified flow diagram of a typical standby system is shown in Fig. 3. Table 8 shows the variables associated with all considered element types of failures.

Assume that the unavailability of the system is the aim of the analysis. The operating policy is as follows:

(1) The system is designed to supply water from tank *T*1 to collector *C*1.
(2) The system must pump using *B*1 and open the motorized valves *V*2, *V*5 and *V*6.
(3) All valves are shown in Fig. 3 in their normal positions (standby system).
(4) Pump *B*1 is checked once a month; during the test, the pump is started and works for 10 minutes, making the water to flow through the manual valve *V*7 and the retention valve *VR*1, after opening the motorized valve *V*2 returning the water to tank *T*1.

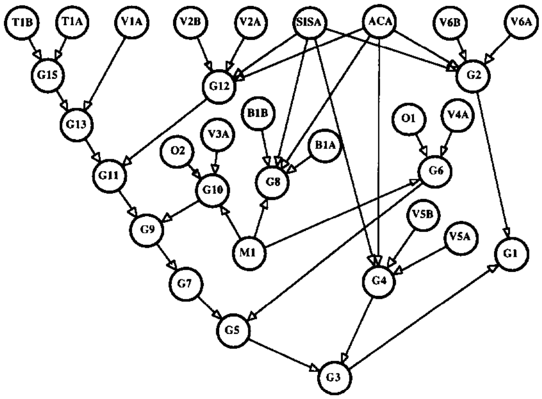

Fig. 4. Directed acyclic graph used to modelize the standby system.

(5) The system works correctly even though valve $V7$ be open.

(6) The maintenance to pump $B1$ is done once every five months. To this aim, valves $V3$ and $V4$ are closed and we assume that the system is unavailable if demand is required during the maintenance period, which takes 7 hours.

(7) No other contributions to unavailability are considered.

(8) In each recharge the availability of the system is checked.

(9) Pump $B1$ must work for 24 hours in order to mitigate one accident.

(10) The power supply of the pump and all motorized valves comes from train A, which has an estimated unavailability of 1.0×10^{-3}. Its failure probability during 24 hours is assumed negligible.

(11) Logical signals of the pump and motorized valves come from train A, which has an estimated unavailability of 1.0×10^{-4}.

(12) Valves $V5$ and $V6$ are tested once a month.

To avoid replication of nodes, as it is usually done with fault tree diagrams, the dependence structure of the involved variables has been modelized by means of the graph in Fig. 4 and its corresponding Bayesian network.

We are interested in obtaining a confidence interval for the probability of failure of the system. In particular, we want to analyze the influence of the failure probabilities (unavailabilities) of the logic signal failure (SISA), the electric power system (ACA) and the maintenance policy (M1) on the probability of failure of the system. It can be shown that the probability of failure of the system can be written as

$$Z = P(G1 = 1) = h(x_1, x_2, x_3) = 1 - \alpha x_1 x_2 x_3, \tag{58}$$

where x_1, x_2 and x_3 are the probabilities of no failure associated with variables $SISA$, ACA and $M1$, respectively, and α is a constant, which is close to 1 and depends on the failure probabilities of the remaining elements in the system.

Since Expression (58) involves the product $x_1x_2x_3$, we can use the results of the previous examples to this case, we can solve the uncertainty analysis problem in two different ways:

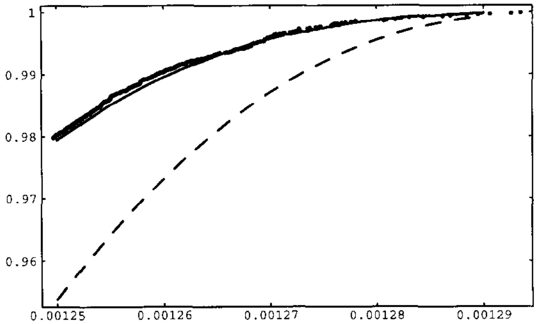

Fig. 5. Exact (continuous line), simulated and FPI (dotted line) tail of $h(x_1, x_2, x_3) = 1 - \alpha x_1 x_2 x_3$ for $\alpha = 0.999$ when $X_i, i = 1,2,3$ are uniforms $U(0.9999, 1)$.

Exact analysis. Expression (58) shows that the cdf of Z is

$$F_Z(z) = \text{Prob}[Z \leqslant z] = \text{Prob}[1 - \alpha x_1 x_2 x_3 \leqslant z] = 1 - F_U[(1-z)/\alpha], \tag{59}$$

where $F_U(u)$ is the cdf of $U = x_1 x_2 x_3$.

If we assume that X_1, X_2 and X_3 are i.i.d. uniform $U(\beta, 1)$ random variables, the cumulative distribution function of U in the region $u < \beta^2$ is

$$F_U(u) = \frac{-\beta^3 + u + u\log\beta + u(\log\beta)^2/2 + u(\log u/\beta^2)^2/2 - (u + u\log\beta)\log(u/\beta^2)}{(1-\beta)^3}. \tag{60}$$

Approximate analysis. It is clear that Z takes values in the interval $(1-\alpha, 1-\alpha\beta^3)$. Using the techniques described in Section 3, we simulate sequentially, X_1, X_2 and X_3 in the intervals

$$L_i = \beta \leqslant X_i \leqslant \frac{\alpha\beta^3 + \varepsilon}{\alpha\beta^2 \prod_{j=1}^{i-1}(x_i/\beta)} = U_1, \quad i = 1,2,3. \tag{61}$$

Fig. 5 shows the exact, the simulated and the FPI tail for $\alpha = 0.999$, $\beta = 0.9999$ and $\varepsilon = 0.00005$. A sample size of $n = 1000$ has been used for the proposed method. Note that the FPI method gives worse results than the proposed method.

From Fig. 5, the 0.98 one-sided confidence interval for the probability of failure of the system is $(0, 0.00125)$. The corresponding interval obtained using the FPI method is $(0, 0.001264)$.

7.2. Example 2: Pressure tank

Here we return to the pressure tank example given in the introduction. Fig. 6 shows the failure tree and the sets of failures that produce system failure. Note that failure of the tank K is equal to the logical expression

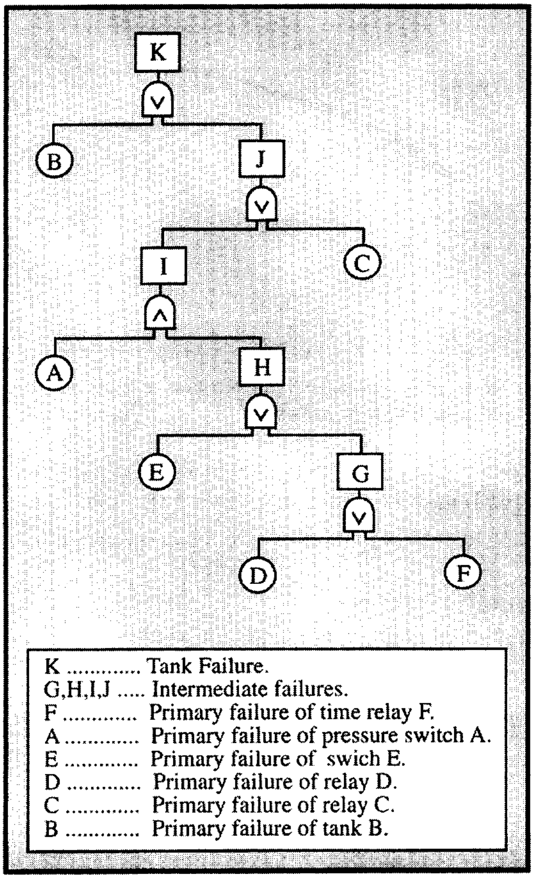
K
∨
B
J
∨
I
C
∧
A
H
∨
E
G
∨
D
F
K .... Tank Failure.
G,H,I,J ..... Intermediate failures.
F .... Primary failure of time relay F.
A .... Primary failure of pressure switch A.
E .... Primary failure of swich E.
D .... Primary failure of relay D.
C .... Primary failure of relay C.
B .... Primary failure of tank B.

Fig. 6. Tank failure tree and logic equation of failure.

$$K = B \vee C \vee (A \wedge E) \vee (A \wedge D) \vee (A \wedge F), \tag{62}$$

where the symbols "∨" and "∧" are used for "*or*" and "*and*", respectively. From (62) we get

$$\begin{aligned} \bar{K} &= \overline{B \vee C \vee (A \wedge E) \vee (A \wedge D) \vee (A \wedge F)} \\ &= \bar{B} \wedge \bar{C} \wedge \overline{(A \wedge E)} \wedge \overline{(A \wedge D)} \wedge \overline{(A \wedge F)} \\ &= \bar{B} \wedge \bar{C} \wedge (\bar{A} \vee \bar{E}) \wedge (\bar{A} \vee \bar{D}) \wedge (\bar{A} \vee \bar{F}) \\ &= (\bar{B} \wedge \bar{C} \wedge \bar{A}) \vee (\bar{B} \wedge \bar{C} \wedge \bar{E} \wedge \bar{D} \wedge \bar{F}). \end{aligned} \tag{63}$$

Eqs. (62) and (63) are the bases for deriving the set of rules for a deterministic expert system. The variables in the set

$$V = \{A, B, C, D, E, F, G, H, I, J, K\},$$

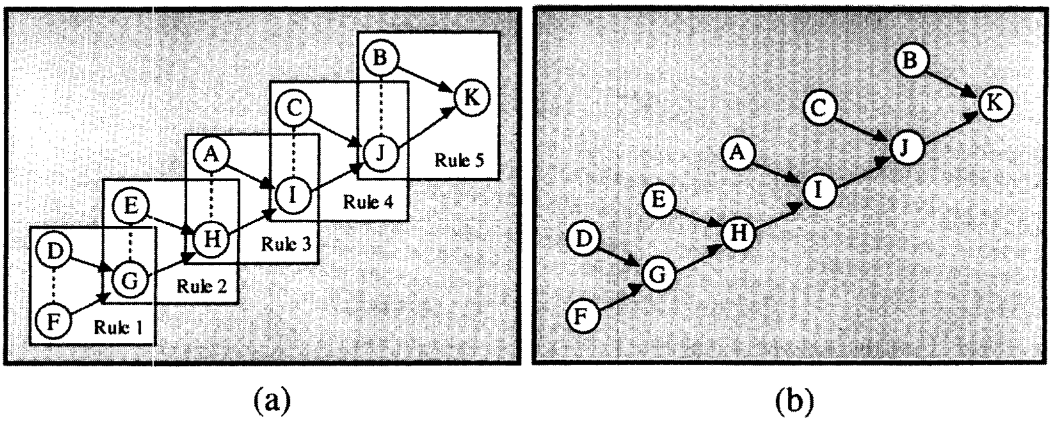

Fig. 7. (a) Sets of chained rules and (b) influence diagram of the pressure tank example.

where G, H, I, and J are intermediate failures, are assumed to be either true or false. Fig. 7(a) shows the set of chained rules as seen in uncertainty paradigms.

Fig. 7(b) shows the directed acyclic graph for the pressure tank system problem, where the arrows indicate the initial dependence of the variables. From Fig. 7(b), the joint probability distribution $P(X)$ of all nodes can be written as

$$\begin{aligned}&P(A)P(B)P(C)P(D)P(E)P(F)\\&\times P(G \mid D,F)P(H \mid E,G)P(I \mid A,H)P(J \mid C,I)P(K \mid B,J),\end{aligned}$$

where $P(A=1)=x_3$, $P(B=1)=x_1$, $P(C=1)=x_2$, $P(D=1)=x_5$, $P(E=1)=x_4$ and $P(F=1)=x_6$, $P(G \mid D,F), P(H \mid E,G), P(J \mid C,I) P(K \mid B,J)$ are the probabilities associated with "OR" gates and $P(I \mid A,H)$ is the probability associated with an "AND" gate. With this notation, the probability of failure becomes

$$\begin{aligned}P(K=1) = {}&x_1 + x_2 - x_1x_2 + x_3x_4 - x_1x_3x_4 - x_2x_3x_4 + x_1x_2x_3x_4\\&+ x_3x_5 - x_1x_3x_5 - x_2x_3x_5 + x_1x_2x_3x_5 - x_3x_4x_5 + x_1x_3x_4x_5\\&+ x_2x_3x_4x_5 - x_1x_2x_3x_4x_5 + x_3x_6 - x_1x_3x_6 - x_2x_3x_6\\&+ x_1x_2x_3x_6 - x_3x_4x_6 + x_1x_3x_4x_6 + x_2x_3x_4x_6 - x_1x_2x_3x_4x_6\\&- x_3x_5x_6 + x_1x_3x_5x_6 + x_2x_3x_5x_6 - x_1x_2x_3x_5x_6 + x_3x_4x_5x_6\\&-x_1x_3x_4x_5x_6 - x_2x_3x_4x_5x_6 + x_1x_2x_3x_4x_5x_6. \qquad (64)\end{aligned}$$

To evaluate the probability of $K=1$ we assume very small probabilities of events A, B, C, D, E and F and then, taking into consideration only terms up to second order we get the approximation given in Expression (1).

Eq. (1) has been used to simulate the tail of $P(K=1)$ assuming the random variables A to F to be uniformly $U(0, 0.001)$ and independently distributed. Fig. 8 shows the right tail approximations of the probability of failure of the pressure tank (without common

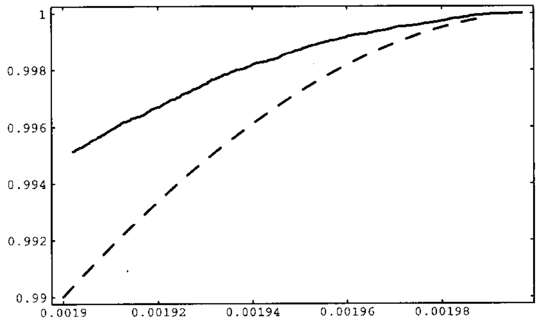

Fig. 8. Right tail approximations of the probability of failure of the pressure tank (without common cause analysis), obtained by the proposed (continuous line) and the FPI (dotted line) methods.

cause analysis), obtained by the proposed (continuous line) and the FPI (dotted line) methods. Note that though the difference is small in terms of probability values, it is of a certain relevance for the tail. Since the proposed method has been shown to be more reliable than the FPI, we can make statements of the form "The probability of the failure probability of the pressure tank to be larger than 0.0019 is less than 0.0052". In other words, we can give unilateral confidence intervals for the failure probability of the pressure tank.

7.2.1. Common cause of failure

Now let us assume that there is a common cause of failure for all relays (C, D and F). So, we draw new edges linking them. Thus, the process can be modified as follows. Fig. 9 shows a Bayesian network which corresponds to the new influence diagram where C, D and F have been linked. Note that, in this case, we have a multiply-connected graph. Assume that the new nodes can also take values "true" or "false". The joint probability function of all nodes can be written as

$$\begin{aligned} P(X) = {} & P(A)P(B)P(C \mid D)P(D)P(E)P(F \mid C,D)P(G \mid D,F) \\ & \times P(H \mid E,G)P(I \mid A,H)P(J \mid C,I)P(K \mid B,J). \end{aligned} \tag{65}$$

The conditional probabilities are as before with the only differences being

$$\begin{aligned} &P(F=1 \mid C=0,D=0) = x_6/4, && P(F=1 \mid C=0,D=1) = 2x_6, \\ &P(F=1 \mid C=1,D=0) = 2x_6, && P(F=1 \mid C=1,D=1) = 4x_6 \\ &P(C=1 \mid D=0) = x_2/2, && P(C=1 \mid D=1) = 3x_2. \end{aligned}$$

This shows that the probability of failure of relay F when relays C and D fail are larger than when they are not and the probability of failure of relay C when relay D fails is larger then when it does not.

Note that these probabilities imply a substantial improvement of relay F (its probability of failure has been reduced approximately to $x_6/4$), because the term $P(F = 1 \mid C = 0, D = 0)$ dominates in the expression $P(F = 1) = \sum_{c,d} P(F = 1 \mid C = c, D =$

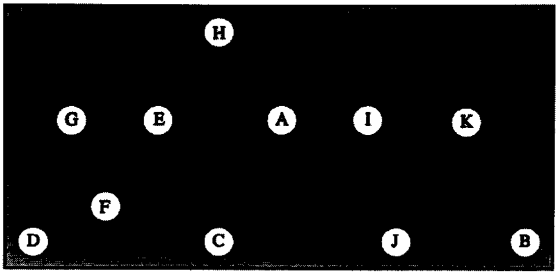

Fig. 9. The Bayesian network for the pressure tank with common causes of failure.

$d)P(C = c, D = d)$ due to the fact that the reliabilities of relays C and D are high. We shall see however that the common causes are close to compensate this effect.

Using the symbolic propagation methods described by Castillo et al. [5] we get the exact expression for the probability of failure of the pressure tank:

$$\begin{aligned}P(K = 1) = {} & x_1 + 0.5x_2 - 0.5x_1x_2 + x_3x_4 - x_1x_3x_4 - 0.5x_2x_3x_4 \\ & + 0.5x_1x_2x_3x_4 + 2.5x_2x_5 - 2.5x_1x_2x_5 + x_3x_5 - x_1x_3x_5 \\ & - 3x_2x_3x_5 + 3x_1x_2x_3x_5 - x_3x_4x_5 + x_1x_3x_4x_5 + 0.5x_2x_3x_4x_5 \\ & - 0.5x_1x_2x_3x_4x_5 + 0.25x_3x_6 - 0.25x_1x_3x_6 - 0.125x_2x_3x_6 \\ & + 0.125x_1x_2x_3x_6 - 0.25x_3x_4x_6 + 0.25x_1x_3x_4x_6 + 0.125x_2x_3x_4x_6 \\ & - 0.125x_1x_2x_3x_4x_6 - 0.25x_3x_5x_6 + 0.25x_1x_3x_5x_6 + 0.125x_2x_3x_5x_6 \\ & - 0.125x_1x_2x_3x_5x_6 + 0.25x_3x_4x_5x_6 - 0.25x_1x_3x_4x_5x_6 \\ & - 0.125x_2x_3x_4x_5x6 + 0.125x_1x_2x_3x_4x_5x_6 \qquad (66)\end{aligned}$$

Taking now into consideration only terms up to second order we get the approximation given in Expression (2), which has been used to simulate the tail of $P(K = 1)$ assuming the random variables A to F to be uniformly $U(0, 0.001)$ and independently distributed. The methods in Section 3 have been used. Fig. 10 shows the simulated tails for the probability of failure of the pressure tank (Common cause analysis), using the proposed method (continuous line) and the FPI method (dotted line). The resulting tail is very similar to the previous tail (see Fig. 8), even though the reliability of relay F has been increased.

8. Conclusions

One efficient computational algorithm for simulating the left (right) tail of a random variable which is defined as an increasing (decreasing) invertible in each variable function of a set of basic variables has been given. This method allows simulating the tail directly, i.e., all the simulated points are guaranteed to belong to the target tail; this leads to a good performance of the method. Several theoretical examples and two

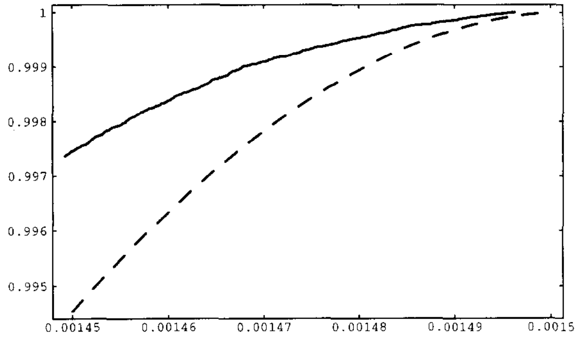

Fig. 10. Simulated tails for the probability of failure of the pressure tank (Common cause analysis) using the proposed method (continuous line) and the FPI method (dotted line).

real life examples have been given to illustrate the method. Some comparisons of the exact cdf in the tail and the FPI method with the simulated cdf shows that the proposed method performs well especially if the simulation method is carefully selected for getting similar scores for all simulated instantiations. The method has immediate applications in many fields of reliability theory, as probability risk or security assessments of complex systems.

Acknowledgments

We thank Iberdrola, the Leonardo Torres Quevedo Foundation of the University of Cantabria and Dirección General de Investigación Científica y Técnica (DGICYT) (project TIC96-0580), for partial support of this work. We also thank the Editor and referees for their comments which allowed a substantial improvement of the paper.

References

[1] B. Arnold, E. Castillo and J.M. Sarabia, Conditionally specified distributions, in: Lecture Notes in Statistics, Vol. 73 (Springer, New York, 1992) 1–151.

[2] R. Bouckaert, E. Castillo and J.M. Gutiérrez, A modified simulation scheme for inference in Bayesian networks, *Internat. J. Approximate Reasoning* (1995).

[3] K. Breitung, Asymptotic approximations for multinormal integrals, *J. Engineering Mechanics Division* ASCE 100 (3) (1984) 357–366.

[4] E. Castillo, *Extreme Value Theory in Engineering* (Academic Press, New York, 1988).

[5] E. Castillo, J.M. Gutiérrez and A.S. Hadi, Parametric structure of probabilities in Bayesian networks, in: *Proceedings ECSQARU'95*, Fribourg, Lecture Notes in Artificial Intelligence, Vol. 946 (Springer, New York, 1995) 89–98.

[6] E. Castillo, J.M. Gutiérrez and A.S. Hadi, *Expert Systems and Probabilistic Network Models* (Springer, New York, 1997).

[7] R.M. Chavez and G.F. Cooper, A randomized approximation algorithm for probabilistic inference on Bayesian belief networks, *Networks* 20 (1990) 661–685.

[8] L. Devroye, *Non-Uniform Random Variate Generations* (Springer, New York, 1986).

[9] A.N. Freudenthal, Safety and the probability of structural failure, *Trans. ASCE* 121 (1956) 1337–1397.

[10] J. Galambos, *The Asymptotic Theory of Extreme Order Statistics* (Krieger, Malabar, FL, 1987).

[11] F.E. Haskin, B.D. Staple and C. Ding, Efficient uncertainty analyses using fast probability integration, *Nuclear Engineering and Design* (1996) 225–248.

[12] A.M. Hasofer and N.C. Lind, Exact and invariant second moment code format, *J. Engineering Mechanics Division* ASCE 100 (1) (1974) 111–121.

[13] J. Pearl, Evidential reasoning using stochastic simulation of causal models, *Artificial Intelligence* 32 (1987) 245–257.

[14] J. Pickands III, Statistical inference using extreme order statistics, *Ann. Statist.* 75 (1) (1975) 119–131.

[15] R. Rackwitz and B. Fiessler, Structural reliability under combined load sequences, *J. Comput. Struct.* 9 (1978) 489–484.

[16] CRY. Rubinstein, *Simulation and the Monte Carlo Method* (John Wiley and Sons, New York, 1981).

[17] R.D. Shachter and M.A. Peot, Simulation approaches to general probabilistic inference on belief networks, in: L.N. Kanal, M. Henrion, R.D. Shachter and J.F. Lemmer, eds., *Uncertainty in Artificial Intelligence*, Vol. 5 (North-Holland, Amsterdam, 1990).

[18] B. Staple and F.E. Haskin, Analysis of extreme top event frequency percentiles based on fast probability integration, in: *Proceedings PSAM-II*, San Diego, CA (1994) 4619–4624.

[19] I. Weissman, Estimation of parameters and large quantiles based on the *k* largest observations, *J. Amer. Statist. Assoc.* 73 (364) (1978) 812–815.