An. Şt. Univ. Ovidius Constanţa Vol. 17(2), 2009, 155–166

COMPUTING THE RELIABILITY OF SYSTEMS WITH STATISTICAL DEPENDENT ELEMENTS

Doina Perpelea

Abstract

Starting from the structure function of a coherent system S, under the disjunctive-conjunctive form, $\Phi(x_1, x_2, ..., x_k) = \overset{m}{\underset{j=1}{\vee}} \overset{l_j}{\underset{i=1}{\wedge}} x_{j_i}$and the vector's probability distribution $T = (T_1, ..., T_k)$ that represents the system's elements reliability or the distributions of the variables T_i conditioned by the variables T_j, $j \neq i$, one determines a procedure of estimating the variable's probabilities $T_S = \max\limits_{j=\overline{1,m}} \{ \min\limits_{i\in\{j_1,...,j_{l_j}\}}\{T_i\}\}$ which represent the reliability of the system. The procedure is based on the Metropolis-Hastings or Gibbs algorithms for the generation of selection values of vector T. One also proposes a method of approximate generation of the random variables with continuous and limited density, a useful method in generating the Gibbs selections.

1 Introduction

The mathematical description of the reliability of a system can be carried out on a global level, ignoring the structure of the system, or on the structural level, taking into consideration the elements of the system and the relations among them (see [1], [3], [4], [5]). The reliability study on the structural level aims at establishing a relationship between the reliability of the system and the reliability of its elements. In the reliability statistical study, the functioning

Key Words: reliability function; estimate; Monte Carlo methods; Markov chains.
Mathematics Subject Classification: 60J10, 62M05, 65C05
Received: March, 2009
Accepted: September, 2009

durations of the system's elements, or the system functioning duration, are described by random variables that are denominated as *functions of reliability.* Further on, we consider a system with k elements and we note with $T_i(i = 1, 2, ..., k)$ the reliability function of ith component and with T_S the reliability function of the system S. One associates to each component $i, i = 1, 2, ..., k$ a binary variable x_i, so that:

$$x_i = \begin{cases} 1 & \text{if component } i \text{ works} \\ 0 & \text{if component } i \text{ is defective,} \end{cases}$$

for $i = 1, 2, ..., k$.
Similarly, we associate to the system a binary variable S (state variable) as it follows:

$$S = \begin{cases} 1 & \text{if the system works} \\ 0 & \text{if the system is defective} \end{cases}$$

With the above notations, the relation between the system state and the components of the system can be expressed by a Boolean function, that we call the structure function, thus:

$$S = \Phi(x_1, x_2, ..., x_k). \tag{1}$$

Similarly one can define a functional relation between the reliability function T_S of the system and the reliability functions $T_i(i = 1, ..., k)$ of the components:

$$T_S = \xi(T_1, ..., T_k). \tag{2}$$

A system whose components are related in series, which, from the reliability point of view means that the system works if and only if all its components work, has the structure function $\Phi(x) = x_1 \wedge x_2 \wedge ... \wedge x_k$, $(x = (x_1, x_2, ..., x_k))$, and the reliability function $T_S = \min(T_1, ..., T_k)$. A system whose components are related in parallel, which, from the reliability point of view means that the system works if and only if one of its components works, has the function of structure $\Phi(x) = x_1 \vee x_2 \vee ... \vee x_k$, and the reliability function $T_S = \max(T_1, ..., T_k)$.

One calls *route,* a vector x, $x = (x_1, x_2, ..., x_k)$ for which $\Phi(x) = 1$ and it corresponds to crowd $C(x) = \{i, x_i = 1\}$. A *route* x is *minimal* if whatever would be $y < x$ we have $\Phi(y) = 0$, where $y < x$ means $y_i \leq x_i, i = 1, 2, ..., k$ cu $y_i < x_i$ for a certain i. Physically, *a minimal route* is a minimal crowd of elements whose functioning assures the functioning of the system.

The structure function is under the disjunctive-conjunctive form if

$$\Phi(x_1, x_2, ..., x_k) = \overset{m}{\underset{j=1}{\vee}} \overset{l_j}{\underset{i=1}{\wedge}} x_{j_i}, \tag{3}$$

where conjunctions

$$\overset{l_j}{\underset{i=1}{\wedge}} x_{j_i}, j = 1, ..., m$$

are determined by crowds $C(x)$ of the minimal routes.

In other words, the system is functioning if there is $j \in \{1, ..., m\}$ and $x_{j_i} = 1$, $i \in \{1, ..., l_j\}$. Therefore, the system functions if for at least a minimal route x all the elements x_i, $i \in C(x)$, function.

If the random variable T_i represents the functioning time of the element i, $i = 1, 2, ..., k$, then out of the relation (3) it results that, on the moment t, the system works if

$$\max_{j=\overline{1,m}} \{ \min_{i \in \{j_1, ..., j_{l_j}\}} \{T_i\}\} > t\,. \tag{4}$$

Thus, the random variable $T_S = \max_j \{\min_i \{T_i\}\}$ represents the functioning time of the system and on moment t

$$\Phi(x_1, ..., x_k) = \begin{cases} 1 & \text{if} \quad T_S > t \\ 0 & \text{if} \quad T_S \leq t. \end{cases}$$

The independence hypothesis of the random variables T_i leads to models for which the probability calculus $P(T_S > t)$, that represents the system reliability, depending on the probabilities $P(T_i > t)$, $i = 1, 2, ..., k$, is calculated according to well-defined procedures depending on the system structure function and the probabilities of the reliability functions of the elements (see [1], [3], [5]). Unfortunately there is a large range of systems for which the independence hypothesis within the defecting time does not concord with reality, the defecting time of an element being dependent on the functioning or non-functioning of other elements. For example, for elements related in parallel and that perform the same task, the falling of one of them may lead to overload for the others and shortens the defecting time.

In general, the probability calculus of variable T_S defined by relation (4) through analytical means becomes complicated because both of the dependency structure between the variables of $T = (T_1, ..., T_k)$ and the structure function Φ.

Monte Carlo methods offer a solution of estimating the probabilities of T_S when one knows the repartition of vector T or the conditioned repartitions of components of T.

What is new in this paper is the application of Markov Chain Monte Carlo methods (MCMC) for computing the reliability systems with statistic dependent elements. For reliability distributions having characteristic proprieties (continuous and bounded on the positive real numbers) the Metropolis algorithms used in the general case, (see [7], [14]) are adjusted for those target distributions (see Section 3, Algorithm 2). In this case it is established a proposed distribution which is valid for all reliability distributions and it is demonstrated the correctness of the algorithms for this distribution (see Section 3, Proposition 1). In the practical cases it is easier to establish the density of T_i for an element i, conditioned by the rest of the elements and Gibbs algorithms (see [7], [14]) if there is a procedure to obtain samples from conditional distributions. The Theorem 2, Section 4, is useful to construct the Algorithm 5 which in combination with Gibbs algorithms makes possibly the employment of Gibbs algorithm in general case. The Theorem 1, Section 2, proves the correctness of the Algorithm 1 which constructs estimations for reliability function with specified accuracy. In the exposed example are compared the histograms for the samples from normal distribution generated with rejection algorithm (see [10], [11]) and Algorithm 5, Section 4. It is also exposed the result of the T test.

2 Reliability calculus through Monte Carlo methods

One considers ξ the real function defined on D^k, with D the positive real axis, according to the law

$$\xi(t_1, ..., t_k) \overset{def}{=} \max_{j=\overline{1,m}} \{ \min_{i \in \{j_1,...,j_{l_j}\}} \{t_i\}\} \ , t_i \in D, i = 1, ..., k, \tag{5}$$

and $(t_1^s, ..., t_k^s)$, $s = 1, 2, ..., n$, are n Bernoulli selection values on the random vector $T = (T_1, ..., T_k)$. It results that the value $t_s^* = \xi(t_1^s, ..., t_k^s)$, $s = 1, 2, ..., n$, represent a Bernoulli selection of volume n on the random variable T_S which gives the defecting time of the system.

For $p(t) = P(T_S > t)$, the probability of functioning of the system on moment t, $t > 0$, the statistics

$$\hat{p}(t) = \frac{1}{n} \sum_{s=1}^{n} \lambda_{(t,\infty)}(t_s^*), \tag{6}$$

with λ_A as the indicator function, represents an unbiased estimate of the parameter $p(t)$ (see [8], 14]); moreover the variance of $\hat{p}(t), D^2(\hat{p}(t))$ is given by the relation

$$D^2(\hat{p}(t)) = \frac{1}{n}p(t)(1-p(t)) = \frac{1}{n}p(t)\cdot q(t), \tag{7}$$

i.e. the product between the defecting probability and the non-defecting probability divided by the volume of selection.

In the hypothesis that one knows a procedure of generating the Bernoulli selection values $t_s = (t_1^s, ..., t_k^s)$ using the repartition of the vector $T = (T_1, ..., T_k)$, it results the following algorithm of approximate calculus of the value $p(t)$ that represents the probability that the system functions on the moment t.

Algorithm 1 *1.* ***Initial*** $n := 0$*;* $\hat{p} := 0$*;* $K = 1/(\varepsilon^2 \cdot \alpha)$ */*α *a small positive real number and* α *a small probability / ;*

2. ***As long as*** $n \leq K$ ***is achieved*** $n := n+1$*; generation of* $(t_1^n, ..., t_k^n)$*, independently of previous generations;* ***one calculates***

$$t_n^* = \xi(t_1^n, ..., t_k^n);$$

$$\hat{p} := \hat{p} + \lambda_{(t,\infty)}(t_n^*);$$

3. ***Output*** $\hat{p} := \hat{p}/n$*.*

Theorem 1 *Algorithm 1 calculates an estimate of* $p(t)$ *with a smaller error than* ε *on a safety coefficient of* $1-\alpha$*.*

Proof. From the Steps 1 and 2 of the Algorithm 1, it results that

$$\hat{p} = \sum_{s=1}^{n} \lambda_{(t,\infty)}(t_s^*). \tag{8}$$

The probability with which $\lambda_{(t,\infty)}(t_s^*)$ equals 1 is

$$P(\lambda_{(t,\infty)}(t_s^*) = 1) = P(T_S > t) = p(t). \tag{9}$$

Out of the variables' independence $t_1^*, ..., t_n^*$ and identically their repartition, it results that $\sum_{s=1}^{n} \lambda_{(t,\infty)}(t_s^*)$ is a random variable distributed binomially by parameters n and $p(t)$.

So, $P(\sum_{s=1}^{n}\lambda_{(t,\infty)}(t_s^*)=k)=C_n^k p(t)^k q(t)^{n-k}$, from where the variance is

$$M(\sum_{s=1}^{n}\lambda_{(t,\infty)}(t_s^*))=np(t)$$

and the dispersion is $D^2(\sum_{s=1}^{n}\lambda_{(t,\infty)}(t_s^*))=np(t)q(t)$
After achieving Step 3, $\hat{p}=\frac{1}{n}\sum_{s=1}^{n}\lambda_{(t,\infty)}(t_s^*)$ and $M(\hat{p})=p(t)$,

$$D^2(\hat{p})=\frac{p(t)q(t)}{n}<\frac{1}{n}.$$

Out of the inequality of Markov (see [8])

$$P(|\hat{p}-p(t)|\leq\varepsilon)\geq 1-\frac{p(t)q(t)}{\varepsilon^2 n}\geq 1-\frac{1}{\varepsilon^2 n}. \tag{10}$$

Out of Step 2, $n\geq\frac{1}{\varepsilon^2\alpha}$, so $\frac{1}{\varepsilon^2 n}\leq\alpha$.
But $1-\frac{1}{\varepsilon^2 n}\geq 1-\alpha$ and relation (10) it results $P(|\hat{p}-p(t)|\leq\varepsilon)\geq 1-\alpha$.

Remark 1 *Algorithm 1 determines a value of n covering through the increase of dispersion $D^2(\hat{p})$ with $\frac{1}{n}$, $p(t)q(t)<1$. A smaller value of n in order to determine the same precision and with the same safe coefficient can be achieved if one approximates $p(t)q(t)$ with $\hat{p}(1-\hat{p})$.*

If $f(x_1,...,x_k)$ is the repartition density of the vector $T=(T_1,...,T_k)$, a general procedure of $(t_1,t_2,...,t_k)$ after the repartition f is the one given by the rejecting-accepting method exposed in [10],[11]. Considering that $T_i\geq 0$, $i=1,2,...,k$, a repartition $h(x_1,...,x_k)$ which wraps up f on the domain D^k, $D=(0,\infty)$, could be the repartition product of k independent random variables with values on the domain $(0,\infty)$. As an example,

$$h(x_1,...,x_k)=\prod_{i=1}^{k}\lambda_i e^{-\lambda_i x_i}, \lambda_i>0, x_i>0, \tag{11}$$

which is the density of k independent random variables distributed exponentially, for which there are a lot of generating algorithms.
The use of this method is conditioned by two aspects

1. Determining a wrapping up h and a constant $M$for which $Mh\geq f$;

2. The number of selection values with repartition h rejected until one accepts one generally increases at the same time with the vector's dimension.

3 Using MCMC methods

Markov Chain Monte Carlo methods (MCMC) offers an extremely generous alternative which is applied to a wide range of target repartitions, repartitions after which one desires to generate random vectors. One build further on a variant of the algorithm Metropolis-Hastings (see [7], [14]) applicable for the definite repartitions on D^k, $D = (0, \infty)$, which admit repartition density as against the Lebesque measure defined on D^k.

One consider D^k with σ-the body defined by the Borelien crowds from D^k and Lebesque measure ℓ_k. On D^k one considers a Markov kernel defined by a density $h(x, y)$ for which there are procedures of selection generation with densities $h(x, .)$ whatever x would be fixed. The repartition defined by $h(x, .)$is called *proposed repartition.*
As an example, $h(x, y) = \prod_{i=1}^{k} \frac{1}{x_i} e^{-y_i/x_i}$, for $x = (x_1, ..., x_k), x_i > 0, i = 1, 2, ..., k$; $y = (y_1, ..., y_k), y_i > 0, i = 1, 2, ..., k$.
One generates a Markov chain with Algorithm 2, chain which has the space of states D^k, is ergodic with invariant (stationary) repartition defined by the density of probability$f$. The repartition defined by $f$is called *target repartition* and represents the repartition according to which one desires the selections obtaining.

Algorithm 2 *1. One chooses arbitrarily an initial value x^1.*

2. Given x^i, one generates x with repartition $h(x^i, \cdot)$.
One puts

$$x^{i+1} = \begin{cases} x & \textit{with probability } \alpha(x^i, x) \overset{def}{=} \frac{f(x)h(x,x^i)}{f(x^i)h(x^i,x)} \wedge 1 \\ x^i & \textit{on the contrary.} \end{cases} \tag{12}$$

Proposition 1 *If the target repartition has the density of probability f positive and continuous on D^k, then the Markov chain generated by Algorithm 2 with the proposed repartition h continuous and positive, is ergodic and has a stationary repartition whose density is f.*

Proof. The nucleus of the chain passing generated by Algorithm 2 is

$$K(x, A) = \int_A \alpha(x, x')h(x, x')\ell_k(dx') + 1_x(A) \int [1 - \alpha(x, x')]h(x, x')\ell_k(dx')$$

with $\alpha(x, x') = \frac{f(x')h(x',x)}{f(x)h(x,x')} \wedge 1$.

The chain is ℓ_k- irreducible (see [7], [9], [14]) because f and h are positive. The chain with the kernel K is reversible and the repartition with density f is invariant. Since f and h are continuous and positive, it results that the chain is aperiodic. Out of irreducibility, invariance of f and aperiodicity, it results that the chain is ergodic (see [2], [9], [12], [14]).

Moreover the Proposition 1, it results the next algorithm of generating the Bernoulli selections of volume n for the random vector $T = (T_1, ..., T_k)$ with the known repartition density f, continuous and positive.

Algorithm 3 *1. One generates Markov Chain $\{x^i\}_{i\geq 0}$ with Algorithm 2 using the target repartition with f density and the h proposed repartition ;*

2. One considers the selection values of vector $T = (T_1, ..., T_k)$ as being $t_s = x^{m_0 s+n_0}$.

In the Algorithm 3, the number n_0 is determined through statistic means and represents the moment when the chain $\{x^i\}_{i\geq 0}$ is considered in the equilibrium state, and m_0 is determined through statistical means so that to determine the independence between $\{t_s\}_{s\geq 0}$.

The statistical determination of the repartition f of the vector $T = (T_1, ..., T_k)$, in many practical cases, becomes difficult, due to the fact that one must monitor the functioning of the system as a whole, eventually on subcrowds of elements which correlate among themselves. Practically, from the statistical point of view it is easier for one to observe the behaviour of an element depending on those elements which could influence its reliability function. Thus, from a statistical point of view for an element i it is much easier for one to determine the function f_i of density of T_i, conditioned by the rest of the elements; considering its dependence on a restrained number of elements. Knowing some selection generation procedures with densities f_i it makes possible the selection generation of vector $T = (T_1, ..., T_k)$ with Gibbs's algorithm (see [7], [14]). Thus, one generates an ergodic Markov Chain having as stationary repartition the repartition of the vector T, after the following algorithm.

Algorithm 4 *Starting from an arbitrary initial state x^1, one brings up-to-date the current state $x^i = (x^i_1, ..., x^i_k)$ to a new state x^{i+1} as it follows.*

For $j = 1, ..., k$: one simulates x^{i+1}_j from $f_j(\cdot/x^{i+1}_1, ..., x^{i+1}_{j-1}, x^i_{j+1}, ..., x^i_k)$.

By applying Theorem 2 from the next section, of approximating the definite continuous functions on $(0, \infty)$ with the average of some Poisson repartitions, the Algorithm 4 becomes an applicable general algorithm, by eliminating the issue of having a generation procedure following the conditioned repartitions.

Practically, when lacking a generation procedure according to density f_i, density f_i is approximated with a mixture of repartitions for which there are simple algorithms to be generated, for example the Algorithm $refal5$ (in the next section).

4 Approximation of continuous functions by medium values

The main objective is to approximate the value of $f(x)$, with x real number, $x > 0$, of a real continuous and bounded function f, through the sum

$$f_n(x) = \sum_{k \geq 0} q_{k,n} \, p_{k,n}(x), \tag{13}$$

where

$$q_{k,n} = \frac{1}{n} f(k/n) \tag{14}$$

and

$$p_{k,n}(x) = n \frac{(nx)^k}{k!} \, e^{-nx} \, . \tag{15}$$

The function $p_{k,n}(x)$ represents the density of a random variable $Z_k = Y_k/n$ and the variable Y_k is an Erlang random variable with parameter $k+1$. Thus, for $\sum_{k \geq 0} q_{k,n} = 1$, the function $f_n(x)$ is the density of a random variable X which is a mixture of random variables Z_k and selections of X are generated on computer through a simple procedure (see [10],[11]).

Theorem 2 *(see [13]) If*

$$f : [0, \infty) \to (-\infty, +\infty)$$

is a continuous and bounded function, then for all x>0,

$$\lim_{n \to \infty} f_n(x) = f(x).$$

For a density function of repartition $f : [0, \infty) \to (0, +\infty)$ there is a real number r so that $\int_r^\infty f(x)dx < \varepsilon$, for a small ε. Then $f(x)$ can be approximated through $f_n^*(x) = \sum_{k=0}^s f(k/n) \frac{(nx)^k}{k!} \, e^{-nx}$ for the integer numbers n and s. Since $\int_0^\infty p_{k,n}(x)dx = \int_0^\infty \frac{y^k}{k!} \, e^{-y} \, dy = 1$ and $\frac{1}{n} \sum_{k=0}^s f(k/n) \approx \int_0^{\frac{s}{n}} f(x)dx$ it results that for $s/n > r$ and $q_k = q_{k,n} / \sum_{k=1}^s q_{k,n}$, $p_k(x) = p_{k,n}(x)$,

$k = 1, \ldots, s$ the function $p(x) = \sum_{k=1}^{s} q_k p_k(x)$ approximates the function $f(x)$ and $p(x)$ is the density of a random variable X.

The following procedure results for the generation of random numbers with the distribution density p:

Algorithm 5 *1. One generates a whole random number k with a probability q_k.*

2. One generates a random number Y_k for the density function of distribution Erlang with parameter $k + 1$.
(Example, $Y_k = -\ln \prod_{i=0}^{k} U_i$ with U_i random numbers uniformly independent on interval $(0, 1)$).

3. Number $X = Y_k/n$ will be a random number with the density function of p distribution.

Example 1 *The histograms for the samples from normal distribution generated with rejection algorithm and Algorithm 5.*

One generated 1000 selection values for the normal standard distribution through a rejection algorithm and Algorithm 4 and one presents further on the tables of frequencies, the histograms and the t statistics probability.

The table of frequency and the histogram of 1000 random values generated by the rejection algorithm.

-3		0
-2,8		3
-2,6		0
-2,4		3
-2,2		4
-2		5
-1,8		15
-1,6		17
-1,4		29
-1,2		35
-1		48

-0,8		52
-0,6		58
-0,4		77
-0,2		75
0		70
0,2		75
0,4		72
0,6		69
0,8		59
1		64
1,2		49

1,4		40
1,6		27
1,8		14
2		15
2,2		13
2,4		5
2,6		3
2,8		2
3		1
		1
		1000

The table of frequency and the histogram of 1000 random values generated by the procedure of approximation.

-3		1
-2,8		0
-2,6		5
-2,4		2
-2,2		3
-2		5
-1,8		13
-1,6		21
-1,4		21
-1,2		26
-1		37
-0,8		57
-0,6		71
-0,4		83
-0,2		86
0		76
0,2		62
0,4		86
0,6		70
0,8		50
1		48
1,2		51
1,4		42
1,6		24
1,8		18
2		16
2,2		8
2,4		7
2,6		6
2,8		1
3		2
		2
		1000

The value of the two queues of the T statistics of the T test of homogeneity, for the generated data is: 0,418013

References

[1] D. J. Andrews, R. T. Moss, *Reliability and Risk Assessment*, Wiley, New York, 1996.

[2] K. B. Athreya, H. Doss, J. Sethuraman, *On the convergence of the Markov chain simulation method*, Ann. Statist., **24** (1996), 69-100.

[3] Gh. Barbu, *Modele de simulare cu aplicaţii în fiabilitate*, Ed. Tehnică, Bucureşti, 1992.

[4] R. E. Barlow, F. Proschan, *Statistical Theory of Reliability and Life Testing*, Holt, Rienhart and Wiston, New York, 1975.

[5] R. Billinton, R. Allan, *Introduction to reliability*, John Wiley and Sons, New York, 1980.

[6] A. A. Borovkov, *Ergodicity and Stability of Stochastics Systems*, Wiley, 1998.

[7] O. Cappe', E. Moulints, T. Ryden, *Inference in Hidden Markov Models*, Springer, 2007.

[8] G. Casella, L. R. Berger, *Statistical Inference*, Duxbury Press, 2001.

[9] W. K. Ching, K. M. Ng, *Markov Chains: Algorithms and Applications*, Springer, 2006.

[10] L. Devroye, *Non Uniform Random Variate Generation*, Springer Verlag, New York, Heidelberg, Tokyo, 1986.

[11] J. Gentle, *Random Numbers Generator and Monte Carlo Methods*, Springer, 1998.

[12] S. P. Meyn, L. R. Tweedie, *Markov Chains and Stochastic Stability*, Springer, 1993.

[13] Gh. Petrescu, *The Approximation of Continuous Functions with Expected Value of Poisson Distribution*, Buletin Stiinţific, Universitatea din Piteşti, Seria Matematică şi Informatică, **12** (2006), 1-8.

[14] C. P. Robert, G. Casella, *Monte Carlo Statistical Methods*, Springer, 2005.

Facultatea de Matematică Informatică,
Universitatea din Piteşti, Romania
E-mail: perpelea_doina58@yahoo.com