EXTRACTA MATHEMATICAE Vol. **19**, Núm. 2, 279 – 288 (2004)

Monte Carlo Simulation of Markov Chain Steady-state Distribution

Abdelaziz Nasroallah

Department of mathematics, Cadi Ayyad University Faculty of Sciences
Semlalia B. P. 2390, *Marrakesh, Morocco*

(Presented by M. Molina)

AMS *Subject Class.* (2000): 60J10, 62G99, 68Q80 *Received May* 14, 2003

1. Introduction

The most widely used mathematical tools to model the behavior of the fault-tolerant computer systems are regenerative Markov processes [1], [2], [10]. Many stationary performance measures of such systems can be written in an explicit form of the stationary distribution of a Markov process. One familiarly form of such measures is $\theta = \sum_{s\in\mathbf{E}} f(s)\pi_s$ where f is a function of state such that $\mathbb{E}[|f(\mathbf{Z})|] = \sum_{s\in\mathbf{E}} |f(s)|\ \pi_s < \infty$, where $\mathbf{Z}$ is an irreducible discrete time Markov chain with a finite state space $\mathbf{E}$, and $\pi = (\pi_s)_{s\in\mathbf{E}}$ is its stationary distribution. For example if $\mathbf{F} \subset \mathbf{E}$ is the subset of nonoperational states of the system, then $\theta = \sum_{s\in\mathbf{E}} \mathbb{I}_{\mathbf{F}}(s)\pi_s$, where $\mathbb{I}_A(x) = 1$ if x is in set A and $\mathbb{I}_A(x) = 0$ otherwise, is the steady-state unavailability of the system. Such measure is studied for a large markovian model in [10].

When analytical computation of θ is very difficult or almost impossible, a Monte Carlo simulation is appealed in order to get estimations. A standard Monte Carlo simulation algorithm (see [9]) fix a regenerative state $s \in \mathbf{E}$ and generate a sample of regenerative cycles from $\mathbf{Z}$: $C_i(s),\ i = 1, \ldots, m$ that start and finish at s, and then use this sample to construct a likelihood estimator of π_s. In this work, we propose a Monte Carlo algorithm that can simultaneously simulate all components π_s of π. The main idea consists on extracting all regenerative cycles corresponding to each state $s \in \mathbf{E}$ from one *long* path of $\mathbf{Z}$. The estimation procedure is justified by the fact that all states can be considered regenerative.

Supported by PSR2001 Cadi Ayyad University.

Since a sample of cycles, used to simulate a component π_s in the standard case, can be viewed as a path containing other regenerative cycles that can be used to simulate other components $\pi_{s'},\ s \neq s'$, then our algorithm, called (VASD), allows economy in simulation time. The path used by VASD must contain all states of $\mathbf{E}$. Furthermore since, for a fixed path, we don't have the same number of cycles for each state, the estimations of π'_ss for which the number of cycles is small can be less consistent. This problem can relatively be solved by considering a path in which the rare state is present moderately. A bootstrapped version of VASD is considered to get consistent estimation from a short path but containing all states. The paper is organized as follows: in the section 2, we describe the Markovian model to simulate. In section 3 , we present the standard Monte Carlo simulation algorithm. In section 4, we propose our algorithm VASD. A bootstrapped version of VASD is examined in section 5. Some numerical illustrations of VASD are proposed in section 6. The last section is concerned with a conclusion.

2. Markovian model to simulate

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space and consider an irreducible continuous time Markov chain $\mathbf{X} = \{X_t\ ,\ t \in \mathbb{R}^+\}$ with finite state space $\mathbf{E}$ and transition rate matrix $A = (a_{i,j})$. We consider the discrete time Markov chain $\mathbf{Z} = \{Z_n\ ,\ n \in I\!N\}$ of $\mathbf{X}$ which transition matrix $P = (p_{i,j})$ is defined by

$$p_{i,j} = \frac{a_{i,j}}{\mid a_{i,i} \mid} \mathbb{I}_{\{i \neq j\}}(i,j), \quad i,j \in \mathbf{E}.$$

Where $|a_{i,i}| := \sum_{i,j \in \mathbf{E}} a_{i,j} \mathbb{I}_{\{i \neq j\}}(i,j)$. Other discretization methods can be found in [8].

Since the process $\mathbf{Z}$ is used to model a fault-tolerant computer system, we assume that $\mathbf{Z}$ starts in the perfect state $i_0 \in \mathbf{E}$ (i.e. all components of the system are operational). It is well known that the simulation of the Markov steady-state distribution does not depend on the started state. Let the sequence (τ_i) defined by

$$\tau_0 = \inf\{n > 0 \mid Z_n = i_0\},$$
$$\tau_i = \inf\{n > \tau_{i-1} \mid Z_n = i_0\}\ ,\ \ i = 1, 2, \dots\ .$$

The process $\mathbf{Z}$ is regenerative and each τ_i is a regeneration point for $\mathbf{Z}$. Now we denote $\pi = (\pi_i)_{i \in \mathbf{E}}$ the stationary probability distribution of $\mathbf{X}$ and consider that a realization or a cycle is a sequence of states which starts at a fixed state,

for example i_0, and finishes when $\mathbf{Z}$ becomes to this fixed state. Since $\mathbf{E}$ is finite and $\mathbf{X}$ is irreducible, the existence of such realizations is well ensured. More details for regenerative Markovian processes can be found in [3, 12, 2, 5]. Our focus is concentrated on the simulation of the stationary performance measures of type $\theta = \mathbb{E}[f(\mathbf{Z})]$ and more generally the stationary probability distribution π. For the existence of θ, we assume that $\mathbb{E}[\mid f(\mathbf{Z}) \mid] < \infty$. To simulate such measures, it is suitable to use the ratio form of θ given by the following proposition (see [9]).

PROPOSITION. *If* $\sum_{j\in\mathbf{E}} \mid f(j) \mid \pi_j < +\infty$ *then*

$$\sum_{j\in\mathbf{E}} f(j)\pi_j = \frac{\mathbb{E}\left[\sum_{k=0}^{\tau_0-1} f(Z_k)h(Z_k)\right]}{\mathbb{E}\left[\sum_{k=0}^{\tau_0-1} h(Z_k)\right]}$$

where $h(i) = 1/|a_{i,i}|$ *is the mean time spent by* $\mathbf{Z}$ *in state* i *and* $(Z_0, \ldots, Z_{\tau_0-1})$ *is the first regenerative cycle of* $\mathbf{Z}$ *relative to the state* i_0.

The proof, which is based on the regenerative property of $\mathbf{Z}$, is given in [4] and in annexe A of [9].

3. Standard Monte Carlo simulation

To simulate θ, the classical algorithms consist on fixing a state, say i_0, and generating a sequence $S_m(i_0) = (C_1(i_0), \ldots, C_m(i_0))$ of m regenerative cycles of $\mathbf{Z}$. Each cycle $C_k(i_0),\ k = 1, \ldots, m$ is in the form

$$C_k(i_0) = \left(Z_0^{(k)}, \ldots, Z_{l_k(i_0)}^{(k)}\right)$$

where $\mathbb{P}[Z_0^{(k)} = i_0] = 1$ and $Z_{l_k(i_0)}^{(k)}$ is the last state generated in the kth cycle before that $\mathbf{Z}$ becomes to the state i_0. So $l_k(i_0)$ is the length of the regenerative cycle $C_k(i_0)$.

Remark. Fixing a state $i_0 \in \mathbf{E}$ can be replaced by generating a state i_0 by a given probability distribution on the state space $\mathbf{E}$. In this case

$$\mathbb{P}[Z_0 = i_0, Z_1 = i_1, \ldots, Z_l = i_l] = \mathbb{P}[Z_0 = i_0] \prod_{j=1}^{l} \mathbb{P}[Z_j = i_j | Z_{j-1} = i_{j-1}].$$

Now, using the last proposition, the measure θ can be estimated by the maximum likelihood ratio

$$r_m(\theta) = \frac{\alpha \sum_{k=1}^{m} G(C_k(i_0))}{(1-\alpha) \sum_{k=1}^{m} H(C_k(i_0))} \tag{1}$$

where $\alpha \in (0,1)$, $G(C_k(i_0)) = \sum_{i=0}^{l_k(i_0)} f(Z_i^{(k)}) h(Z_i^{(k)})$ and $H(C_k(i_0)) = \sum_{i=0}^{l_k(i_0)} h(Z_i^{(k)})$ The classical estimator $r_m(\theta)$ is biased. The bias reduction of the estimator of a ratio is studied in annexe C of [9].

The parameter α in (1) is introduced to control the proportion of realizations, from the total m, to allocate to the numerator and to the denominator. Generally, we take $\alpha = 1/2$, but in some situations, more importance is given to the numerator than to the denominator by taking for example $\alpha = 2/3$. The last choice of α is considered in a variance reduction context, when there are rare events in the simulation (i.e. there are some states which are rarely visited by the process $\mathbf{Z}$, see annexe B of [9]).

In this work, we are concerned with the use of the path $S_m(i_0)$ to get estimations for some parameters. In the classical case, this path is used to simulate a stationary performance of type θ. Our contribution is to use the same path $S_m(i_0)$ not only to simulate θ which is a combination of the components of π but to simulate all the vector π directly. We call our procedure Vectorial Algorithm for Stationary Distribution (VASD).

4. Vectorial algorithm for stationary distribution

The idea of our approach is to look at the sequence of regenerative cycles $S_m(i_0)$ as a path of $\mathbf{Z}$ and for each state $i \in \mathbf{E}$, we consider the sample of regenerative cycles $S_{m_i}(i) = (C_1(i), \ldots, C_{m_i}(i))$, $i \in \mathbf{E}$ extracted from the same path $S_m(i_0)$. It is important to note that the existence of such cycles is guaranteed by the fact that each state of the process $\mathbf{Z}$ is a regenerative point. So while a classical procedure fix a state and generates regenerative cycles relative to this state, our approach is to fix all states and use each cycle related to each state encountered along the path $S_m(i_0)$. This allows to estimate all the components π_i, $i \in \mathbf{E}$ at the same time. For example if $\mathbf{E}4$ has 4 states denoted as $x_1, \ldots, x_4$ and if

$$x_1x_3x_2x_1x_3x_2x_3x_1x_2x_1x_3x_2x_1x_4x_2x_3x_2x_1x_3x_2x_4x_1x_2x_3x_4x_1$$

is a path of $\mathbf{Z}$, the classical procedure, and after fixing state x_1 , considers the following sequence of regenerative cycles

$$S_7(x_1) = (x_1x_3x_2,\ x_1x_3x_2x_3, x_1x_2,\ x_1x_3x_2, \\ x_1x_4x_2x_3x_2,\ x_1x_3x_2x_4,\ x_1x_2x_3x_4)$$

and the estimation of θ is deduced from these 7 cycles. Our algorithm considers, from the same path, the following samples of regenerative cycles

$$S_7(x_1) = (x_1x_3x_2,\ x_1x_3x_2x_3,\ x_1x_2, x_1x_3x_2, \\ x_1x_4x_2x_3x_2,\ x_1x_3x_2x_4,\ x_1x_2x_3x_4),$$
$$S_6(x_3) = (x_3x_2x_1,\ x_3x_2,\ x_3x_1x_2x_1,\ x_3x_2x_1x_4x_2,\ x_3x_2x_1,\ x_3x_2x_4x_1x_2),$$
$$S_7(x_2) = (x_2x_1x_3,\ x_2x_3x_1,\ x_2x_1x_3,\ x_2x_1x_4,\ x_2x_3,\ x_2x_1x_3,\ x_2x_4x_1),$$
$$S_2(x_4) = (x_4x_2x_3x_2x_1x_3x_2,\ x_4x_1x_2x_3).$$

Each sample $S_{m_i}(x_i)$ is then used to simulate π_i, where m_i is the number of regenerative cycles relative to the state x_i.
Now, using the last proposition, a classical estimator of the vector $\pi = (\pi_i)_{i\in\mathbf{E}}$ is then given by

$$r_m(\pi) = (r_{m_i}(\pi_i))_{i\in\mathbf{E}}$$

where

$$r_{m_i}(\pi_i) = \frac{\alpha \sum_{k=1}^{m_i} G(C_k(i))}{(1-\alpha)\sum_{k=1}^{m_i} H(C_k(i))}, \quad i \in \mathbf{E}$$

with $\alpha \in (0,1)$,

$$G(C_k(i)) = \sum_{j=1}^{l_k(i)} \mathbb{I}_{\{i\}}(Z_j^{(k)})h(Z_j^{(k)}) \quad \text{and} \quad H(C_k(i)) = \sum_{j=1}^{l_k(i)} h(Z_j^{(k)})$$

Remarks. 1. If we have an estimation of π, we can then deduce an estimation of any combination of the components $\pi_i,\ i \in \mathbf{E}$ (i.e. we can deduce an estimation of any measure of type θ).

2. The path considered must contain all states. This can be ensured if the path contains some regenerative cycles corresponding to the rare state (i.e. state that have the lower probability to be generated).

Now, for a given path of $\mathbf{Z}$, some states s can have a small number m_s of cycles. Consequently, the estimation of π_s can be poor. To solve this problem, we propose a bootstrapped version BVASD of VASD.

5. Bootstrapped version of vasd

The bootstrap is a resampling technique that approximates the unknown probability distribution $\mathbf{F}$ of the sample $S_{m_i}(i) = (C_1(i), \ldots, C_{m_i}(i))$ by the empirical distribution $\mathbf{F}_{m_i}$ that assign the uniform mass $1/m_i$ to each realization $C_k(i),\ k = 1, \ldots, m_i$ and zero to each cycle which is not in the sample $S_{m_i}(i)$. This technique was initially introduced by Efron in [6] and now is used in a lot of simulation procedures. This nonparametric and empirical method allows, in many situations, to give stable estimators. For example, in the discrimination analysis context, in [7] Efron showed that this technique gives estimators with variance less than the jackknife method. Efron has defined this method as another look at the jackknife one (see [6]). It is well known that the jackknife method is generally used to reduce the estimator bias (see for example annexe C of [9]).

In our context, using this technique, we consider a bootstrapped sample of size m_i generated from the empirical distribution $\mathbf{F}_{m_i}$. We reiterate this procedure to obtain BO bootstrapped samples of size m_i: $S_{m_i}^{(1)}(i), \ldots, S_{m_i}^{(BO)}(i)$, where

$$S_{m_i}^{(b)}(i) = (C_1^{(b)}(i), \ldots, C_{m_i}^{(b)}(i)), \quad b = 1, \ldots, BO$$
$$C_k^{(b)}(i) = (Z_{k,1}^{(b)}, \ldots, Z_{k,l_k(i)}^{(b)}), \quad k = 1, \ldots, m_i, \ \ b = 1, \ldots, BO$$

with $Z_{k,j}^{(b)}$ is the jth state of the kth cycle, relative to the state i, in the bth bootstrapped sample. An estimator of the vector π under the bootstrap technique is given by

$$r_m^{boot}(\pi) = (r_{m_i}^{(boot)}(\pi_i))_{i \in \mathbf{E}},$$

where

$$r_{m_i}^{(boot)}(\pi_i) = \frac{1}{BO} \sum_{b=1}^{BO} \frac{\alpha \sum_{k=1}^{m_i} G(C_k^{(b)}(i))}{(1-\alpha) \sum_{k=1}^{m_i} H(C_k^{(b)}(i))}. \tag{2}$$

Remark that instead of fixing the sizes of bootstrapped samples to m_i for a state i, one can change these sizes in bootstrap iterations.

6. Numerical results

We present two examples of Markovian models to illustrate our algorithm. The first model is a birth and death process of three states and the second model is a Markovian process with 16 states. The first (resp. the second)

model can be used to study the behavior of a fault−tolerant computer system with two (resp. four) components which can be failed and repaired.

Birth and death process

Consider a birth and death process $\mathbf{X}$ which models the behavior of a two components fault−tolerant computer system. The system is in state 1 if the two components are operational, the state 2 corresponds to the fact that only one of the two components is operational, and the system is in state 3 if the two components are non-operational. The state space $\mathbf{E} = \{1, 2, 3\}$, and the transition rate diagram of the process is given in Figure 1.

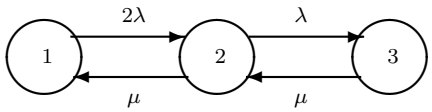

Figure 1 : Transition rate diagram of a birth and death process with 3 states.

The parameters λ and μ are the failure and the repair rates respectively. The transition matrix of $\mathbf{Z}$ is $P = (p_{i,j})_{i,j\in\mathbf{E}}$ where $p_{i,i} = 0$ for $i = 1, 2, 3$; $p_{1,2} = p_{3,2} = 1$ and $p_{2,1} = 1 - p_{2,3} = \mu/(\lambda + \mu)$ and the other coefficients of P are equal to zero. We fix $\mu = 1$ and consider three values of λ: $10^{-1}, 10^{-2}$ and 10^{-3}. For each case, we simulate 105000 events for the process $\mathbf{Z}$, where one event is equivalent to a state transition. The results are summarized in Table 1 as follows: The exact stationary distribution $\pi^{(exa)}$ is given in column3, an estimation $\pi^{(est)}$ of $\pi^{(exa)}$ is given in column 4, in column 5 we find the 99% confidence interval for each component of $\pi^{(est)}$, and the last column is concerned with the number of regenerative cycles used to estimate each component $\pi_i^{(exa)}$. The approximations used for the construction of confidence intervals can be found in [11]. We remark, for $\lambda = 10^{-1}$ for example, that when the classical algorithms use 47673 cycles by fixing the state 1 (the cycles are deduced from the 105000 events) to simulate a combination of the $\pi_i^{(exa)}$'s, VASD uses the same information (105000 events) to get 47673 cycles to estimate $\pi_1^{(exa)}$, 52500 cycles to estimate $\pi_2^{(exa)}$ and 4827 cycles to estimate $\pi_3^{(exa)}$. The total number of cycles extracted from 105000 events is then 105000 cycles. This additional information is then used to estimate all components of $\pi^{(exa)}$ at the same time. The same remark can be done for the cases $\lambda = 10^{-2}$ and $\lambda = 10^{-3}$. The mean relative error for $\lambda = 10^{-1}$ is 0.0048, which is an indication for a good estimation. Note that since $\sum_{i\in\mathbf{E}} \pi_i = 1$, one can only estimate $\mathrm{Card}(\mathbf{E}) - 1$ components.

λ		$\pi^{(exa)}$	$\pi^{(est)}$	99% conf.int.	cycles
10^{-1}	π_1	0.8197	0.8194	0.1011×10^{-2}	47673
	π_2	0.1639	0.1641	0.3848×10^{-3}	52500
	π_3	0.1639×10^{-1}	0.1660×10^{-1}	0.6237×10^{-3}	4827
10^{-2}	π_1	0.9802	0.9802	0.4395×10^{-4}	51975
	π_2	0.1960×10^{-1}	0.1961×10^{-1}	0.2130×10^{-4}	52500
	π_3	0.1960×10^{-3}	0.1985×10^{-3}	0.2092×10^{-4}	525
10^{-3}	π_1	0.9980	0.9980	0.1354×10^{-5}	52452
	π_2	0.1996×10^{-2}	0.1996×10^{-2}	0.6831×10^{-6}	52500
	π_3	0.1996×10^{-5}	0.1862×10^{-5}	0.6293×10^{-6}	48

Table 1: Estimation of π for the model with 3 states.

Fault−tolerant system with 16 states

Consider a fault−tolerant system which is modelled by a continuous time Markov process $\mathbf{X}$, with the transition rate diagram given in Figure 2.

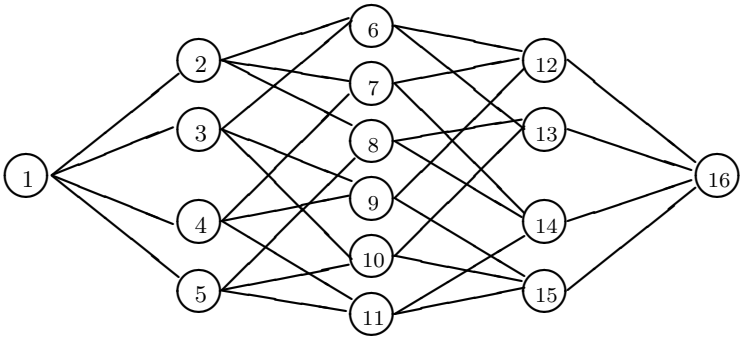

Figure 2 : Transition rate diagram of the model with 16 states.

We have $\mathbf{E} = \{1, \ldots, 16\}$. For two states i and j such that $i < j$, the transition rate from i to j is a failure rate and the transition rate from j to i is a repair rate. We assume that all repair rates are equal to μ and all failure rates between states 1 to 15 are equal to λ_o; the failure rate from states 12, 13, 14, 15 to state 16 are equal to λ_f. For two states which are not related directly by a line in Figure 2, the repair and the failure rates are equal to zero.

We take $\mu = 1$, $\lambda_o = 0.5$ and $\lambda_f = 0.1$, and we simulate 200000 events for this model. The results are given in Table 2 in a similar order of Table 1.

	VASD				BVASD		
	$\pi^{(exa)}$	$\pi^{(est)}$	99% conf.int.	cycles	$\pi^{(boot)}$	$\sigma^{(boot)}$	cycles
π_1	0.199501	0.199527	0.000026	30537	0.197791	0.008218	4485
π_2	0.099751	0.099460	0.001923	19028	0.099341	0.004588	2842
π_3	0.099751	0.101729	0.001930	19462	0.096881	0.003368	2769
π_4	0.099751	0.098657	0.001915	18872	0.096741	0.004272	2831
π_5	0.099751	0.098982	0.001913	18936	0.103449	0.004486	2953
π_6	0.049875	0.051469	0.001354	11816	0.051809	0.002801	1755
π_7	0.049875	0.049351	0.001329	11327	0.049071	0.002291	1681
π_8	0.049875	0.049913	0.001320	11459	0.051972	0.002529	1770
π_9	0.049875	0.049955	0.001360	11464	0.048132	0.002230	1653
π_{10}	0.049875	0.049746	0.001354	11418	0.050781	0.003486	1712
π_{11}	0.049875	0.048975	0.001319	11242	0.052626	0.002822	1813
π_{12}	0.024938	0.024986	0.000926	5927	0.024432	0.001215	867
π_{13}	0.024938	0.025330	0.000970	6009	0.025459	0.001422	907
π_{14}	0.024938	0.024756	0.000933	5872	0.025863	0.001305	911
π_{15}	0.024938	0.024851	0.000945	5894	0.026207	0.001501	920
π_{16}	0.002494	0.002412	0.000237	737	0.002827	0.000144	131

Table 2: Estimation of π for the model with 16 states by VASD and BVASD.

We have the same establishments as for the birth and death model. The classical algorithms use 200000 events to estimate a combination of $\pi_i^{(exa)}$'s when VASD exploits the same information (200000 events) to get 200000 cycles for estimating all the components $\pi_i^{(exa)},\ i = 1, \ldots, 16$ at the same time. Here the mean relative error is 0.0105.

Now we consider 30000 events for the same model (16 states). VASD uses this information to get temporary estimations of the $\pi_i^{(exa)}$'s. Using these estimations as a sample of observations and by applying the bootstrap procedure, we generate 30 bootstrapped samples of sizes 500 for each component and we use the equation (2), to get estimation $\pi^{(boot)}$ of $\pi^{(exa)}$. The bootstrapped simulation results are given in the three last columns of Table 2, where $\sigma^{(boot)}$ is the estimated vector of the standard deviations of components $\pi_i^{(boot)},\ i = 1, \ldots, 16$. The mean relative error is 0.0361. We remark that for a partial information (30000 events), we can get acceptable estimation of $\pi^{(exa)}$ with an additional effort (generating $30 \times 500 \times (\mathrm{Card}(\mathbf{E}) - 1)$ random number in $[0, 1]$) which doesn't depend on the model parameters directly. The exact values of π given in this paper, are computed by a Pentium 4 computer using the Maple V package.

7. Conclusion

We have proposed a vectorial algorithm VASD to simulate the stationary probability distribution π of a regenerative Markov process with finite state space (i.e. vectorial in the sense that it allows to simulate all the vector π at the same time). When classical Monte Carlo algorithms estimate a combination of the components of π by using a path of a Markov chain, VASD exploits the same path to extract additional information to estimate all the vector π directly. Also we have proposed a bootstrapped version, BVASD, of VASD which allows to get more information from a more reduced path (a *short* path) for simulating π. These propositions are more important if we know only a path of the Markov chain and the mean holding time of each state. In this case, VASD and BVASD can be viewed as Monte Carlo approach allowing economy in simulation time.

References

[1] Arlat, J., Méthode et outils pour l'evaluation de la sûreté de fonctionnement des systèmes informatiques tolérant les pannes, in "Technical Report, LAAS, France", 1987, 87 – 336.
[2] Aven, T., Jensen, U., "Stochastic Models in Reliability", Springer-Verlag, New York, 1999.
[3] Crane, M.A., Iglehart, D.L., Simulating stable stochastic systems, III: Regenerative processes and discrete event simulation, *Operation Research,* **23** (1975), 33 – 45.
[4] Chung, K.L., "Markov Chains with Stationary Transition Probabilities", Springer Verlag, Berlin, 1960.
[5] Çinlar, E., "Introduction to Stochastic Processes", Prentice Hall, 1975.
[6] Efron, B., Bootstrap methods: another look at the jackknife, *Annals of Statistics,* **7** (1979), 1 – 26.
[7] Efron, B., Estimating the error rate of a prediction rule: improvement on cross-validation, *J. Amer. Statist. Assoc.,* **78** (1983), 316 – 331.
[8] Hordijk, A., Iglehart, D.L., Schassberger, R., Discrete time methods for simulating continuous time Markov models, *Adv. in Appl. Probab.,* **8** (1976), 772 – 788.
[9] Nasroallah, A., "Simulation de Chaînes de Markov et Techniques de Réduction de la Variance", Doctorat de l'Université de Rennes I, IRISA - France, 1991.
[10] Nasroallah, A., Simulation of high available fault-tolerant systems by simulating stiff and large Markov chains, *Extracta Mathematicae,* **12(1)** (1997), 25 – 31.
[11] Nasroallah, A., Simulation of transient performance measures for stiff Markov chains, *RAIRO Oper. Res.,* **34** (2000), 385 – 396.
[12] Ross, S.M., "Stochastic Processes", John Wiley and Sons, New York, 1983.