PERFORMANCE EVALUATION OF NON MARKOVIAN STOCHASTIC DISCRETE EVENT SYSTEMS - A NEW APPROACH

Serge Haddad, Lynda Mokdad [*,1] **Patrice Moreaux** [**,1]

[*] *LAMSADE, UMR CNRS 7024, Université Paris Dauphine, Place du Maréchal de Lattre de Tassigny, 75775 PARIS, FRANCE*
[**] *LAMSADE & CReSTIC, Université de Reims Champagne-Ardenne, REIMS, FRANCE*

Abstract In this work, we address the problem of transient and steady-state analysis of a stochastic discrete event system which includes non Markovian distributions with a finite support. Rather than computing an approximate distribution of the model (as done in the previous methods), we develop an exact analysis of an approximate model. The design of this method leads to a uniform handling for the computation of the transient and steady-state behaviour of the model. We have evaluated our method on a standard benchmark (the queuing model $M/D/S/K$). Our results demonstrate that: in most of the cases the solution of the approximate model converges quickly to the solution of the exact model, in the difficult cases (e.g. an heavy load on the queue) our method is more robust than the previous ones.

Keywords: Discrete event systems, Stochastic processes, Markovian processes, Periodic processes, Uniformization

1. INTRODUCTION

The transient and steady-state analysis of Markovian discrete event systems is now well established with numerous tools at the disposal of the modellers. The main open issue is the reduction of the space complexity induced by this analysis. However in a realistic system, the distribution of the occurrence (or the duration) of some events can not be described by a exponential law (e.g. the triggering of a time-out). Theoretically any "reasonable" distribution is approximated by a phase-type distribution enabling again a Markovian analysis (Cox 1955*b*). Unfortunately the continuous time Markov chain (CTMC) associated to this approximation is so huge that it forbids its analysis (indeed even its construction). Such a phenomenon often occurs when the non exponential distribution has a finite support (non null dirac, uniform, etc.); then a good phase-type approximation requires too much stages for its specification.

Hence the research has focused on alternative methods. In the case of a single realization of a non Markovian distribution at any time, successful methods have been proposed (German *et al.* 1995) both for the transient and steady state analysis. Let us cite, for instance, the method of supplementary variables (Cox 1955*a*, German and Lindemann 1994) or the method of the subordinated Markov chains (Ajmone Marsan and Chiola 1987).

The general case (i.e. simultaneous multiple realizations of such distributions) is more intricate. The method of supplementary variables is still theoretically applicable but the required space and the computation time limit its use to very small examples. An

[1] {haddad,mokdad,moreaux}@lamsade.dauphine.fr

alternative approach is described for non null Dirac distributions in (Lindemann and Schedler 1996). The stochastic process is observed at periodic moments of time ($\{i\Delta \mid i \in \mathbb{N}\}$) and this new process is expressed by a system of integro-differential equations and solved numerically. The steady-state distributions of these processes are identical and, with another computation, one obtains the transient distribution of the original process from some transient distribution of the transformed process. This method has been implemented in the DSPNexpress tool (Lindemann *et al.* 1999).

Here we propose to define an approximate model on which we perform an exact analysis. Let us first describe the kind of model we will analyse: its behaviour is described by a CTMC, a discrete time Markov chain (DTMC) and a time interval (say Δ). During an interval $(i\Delta,(i+1)\Delta)$ the behaviour is driven by the CTMC and at an instant $i\Delta$ a untimed probabilistic change of state is processed according to the DTMC.

Now we describe how we obtain such a model from a general model with non Markovian finite support distributions. At first, we transform such a distribution into a discrete distribution concentrated on the instants $i\Delta$ included in the support. This step can be done numerically in a straightforward way from the distribution function. So we will not develop it in the remaining paper. Let us note that Δ seems to be the approximation parameter but a better parameter is the ratio between Δ and the size of the support (moreover this indicator is the key factor for the complexity of our analysis).

In the approximate process, the Markovian events occur during the intervals $(i\Delta,(i+1)\Delta)$ and their logical occurrence is specified within the CTMC. Let us note that this is not an approximation since the set $\{i\Delta \mid i \in \mathbb{N}\}$ has a null measure. Non Markovian events always occur in $\{i\Delta \mid i \in \mathbb{N}\}$. Let us describe how they are scheduled. When a non Markovian event is enabled in an interval $(i\Delta,(i+1)\Delta)$ due to the occurrence of a Markovian event then its approximate distribution is interpreted as the number of points $i\Delta$ that must be met before its occurrence. Here we can choose whether we count the next point (an under-evaluation) or not (i.e. an overestimation). The impact of this choice will be discussed later. Thus the residual number of points to be met is included in the state. At any moment $i\Delta$, the current residual numbers are decreased. If some residues are null then the corresponding events occur with possibly some probabilistic choice in case of conflicts. The occurrence of these events may enable new non Markovian events. Such events are handled similarly except that the next point is always counted since now it corresponds to a complete interval.

This process is not exactly ergodic since the steady-state distribution depends on the relative position w.r.t. the points $i\Delta$. The transient analysis is done by successively computing the state distribution at the instants $\Delta, 2\Delta, \ldots$ applying a transient analysis of the CTMC during an interval Δ (via the uniformization technique (Gross and Miller 1984)) followed by a "step" of the DTMC. In order to smooth the effect of the discretization, we average the distribution on the last interval (with a variant of the uniformization). For the steady-state analysis, one computes the steady-state distribution at the instants $i\Delta$ and then starting from this distribution, one averages the steady-state distribution upon an interval.

The balance of the paper is the following one. In section 2 we present our approach. Section 3 reports evaluations of the method applied to a first model. At last, we conclude and give indications on future developments of our work.

2. A NEW APPROXIMATE METHOD

2.1 Definition of the approximate process

As discussed in the introduction, we choose a time interval Δ. We approximate the original stochastic process X with the process Y. Roughly speaking, the approximation is related to the non Markovian events which can only occur at times $t_h = h\Delta$. The impact of this approximation will be evaluated in the next section.

We then study the evolution of the stochastic process Y in each interval (t_h, t_{h+1}) and during the state changes at time t_h. We denote by t_h^- (t_h^+) the "time" before (after) the state change in t_h.

The process Y is defined by two components:

- the subordinated process in (t_h, t_{h+1}) associated to states at t_h^+ records only exponential events. It is then a CTMC defined by its generator $\mathbf{Q}$.
- the state changes at t_h are defined by a stochastic matrix $\mathbf{P}[i,j] = \Pr(X(h\Delta^+) = j \mid X(h\Delta^-) = i)$.

Thus Y is fully defined by its initial probability vector $\pi(0)$ and the matrices $(\mathbf{P},\mathbf{Q})$. These three components depends obviously on Δ since for instance the state space includes the residual number of instants.

2.2 Transient analysis

Let $\pi(h\Delta+\tau)$ the probability vector of the process Y at time $h\Delta+\tau$. We have $\pi(h\Delta+\tau) = \pi(h\Delta^+)e^{\mathbf{Q}\tau}$ for $0 \leq \tau < \Delta$, and $\pi((h+1)\Delta^-) = \pi(h\Delta^+)e^{\mathbf{Q}\Delta}$, so that $\pi((h+1)\Delta^+) = \pi(h\Delta^+)e^{\mathbf{Q}\Delta}\mathbf{P}$ and finally

$$\pi((h+1)\Delta^+) = \pi(0)(e^{\mathbf{Q}\Delta}\mathbf{P})^{h+1} \qquad (1)$$

Since we want to smooth the discretization effect, we define the approximate value $\pi^{(a)}(h\Delta)$ of $\pi_X(h\Delta)$ as the averaged value of the probabilities of the states of

Y in (t_h, t_{h+1}): $\boldsymbol{\pi}^{(a)}(h\Delta) = \frac{1}{\Delta}\int_{h\Delta}^{(h+1)\Delta} \boldsymbol{\pi}(\tau)d\tau$.
Using (1) we have then:

$$\boldsymbol{\pi}^{(a)}(h\Delta) = \frac{1}{\Delta}\boldsymbol{\pi}(0)\,(e^{\mathbf{Q}\Delta}\mathbf{P})^h \int_0^\Delta e^{\mathbf{Q}\tau}d\tau \qquad (2)$$

Finally, we are in general interested by performance measures defined on the states *of the system*, and not on the states of the stochastic process Y. Hence, all components of $\boldsymbol{\pi}^{(a)}(t)$ corresponding to a given state of the original system (i.e. when forgetting the residual numbers) are summed up to compute performance measures (see the example in section 3).

2.3 Steady-state analysis

We also set the approximate value $\boldsymbol{\pi}^{(a)}$ of $\boldsymbol{\pi}$ as the averaged value of the steady-state probabilities of the states of Y over (t_h, t_{h+1}) since this steady-state distribution depends on the relative position in the interval. If $\boldsymbol{\pi}^{(\Delta)}$ is the steady-state distribution of the DTMC with initial distribution $\boldsymbol{\pi}(0)$ and transition probabilities $\mathbf{P}^{(\Delta)} = e^{\mathbf{Q}\Delta}\mathbf{P}$ then

$$\boldsymbol{\pi}^{(a)} = \frac{1}{\Delta}\boldsymbol{\pi}^{(\Delta)} \int_0^\Delta e^{\mathbf{Q}\tau}d\tau \qquad (3)$$

By definition, $\boldsymbol{\pi}^{(\Delta)}$ is the probability vector solution of $\boldsymbol{\pi} = \boldsymbol{\pi}.\mathbf{P}^{(\Delta)}$.

As in the transient case, all components of $\boldsymbol{\pi}^{(a)}$ corresponding to a given state of the system are summed up to compute performance indices.

2.4 Numerical considerations

Formulae (2) and (3) for transient and steady-state probabilities involve vector-matrix products with possibly very large matrices, either $e^{\mathbf{Q}\Delta}$ or $I^{(\Delta)} = \int_0^\Delta e^{\mathbf{Q}\tau}d\tau$. Moreover, it is well known that, although $\mathbf{Q}$ is generally very sparse (see section 3 for an example), $e^{\mathbf{Q}\tau}$ is not sparse (the "fill in" phenomenon). The usual approach to compute these matrices is then based on series expansion and numerical summation until a required precision level. We have implemented a similar algorithm to compute the vector-matrix products directly using the series expansions.

When we need $e^{\mathbf{Q}\tau}$ we follows the uniformization approach (Stewart 1994). If $\mathbf{P}^{(u)} = \mathbf{I} + \frac{1}{u}\mathbf{Q}$ is the uniformized matrix of $\mathbf{Q}$ with rate $u > \max_i\{|q_{ii}|\}$, we have

$$e^{\mathbf{Q}\tau} = \sum_{k\geq 0} e^{-u\tau}\frac{(u\tau)^k}{k!}\left(\mathbf{P}^{(u)}\right)^k \qquad (4)$$

The computation of $I^{(\Delta)} = \int_0^\Delta e^{\mathbf{Q}\tau}d\tau$ is based on (4). By definition,

$$I^{(\Delta)} = \sum_{k\geq 0}\left[\int_0^\Delta e^{-ut}\frac{(ut)^k}{k!}dt\right]\left(\mathbf{P}^{(u)}\right)^k$$

Using an integration by part,

$I_k = \int_0^\Delta e^{-ut}\frac{(ut)^k}{k!}dt$ may be expressed as

$$I_k = -\frac{e^{-u\Delta}(u\Delta)^k}{uk!} + I_{k-1} \quad \text{for } k \geq 1$$

Hence,

$$I_0 = -\frac{e^{-u\Delta}}{u} + \frac{1}{u}$$
$$I_k = \frac{1}{u}\left[1 - e^{-u\Delta}\sum_{h=0}^{h=k}\frac{(u\Delta)^h}{h!}\right]$$

and finally

$$I^{(\Delta)} = \frac{1}{u}\sum_{k\geq 0}\left[1 - e^{-u\Delta}\sum_{h=0}^{h=k}\frac{(u\Delta)^h}{h!}\right]\left(\mathbf{P}^{(u)}\right)^k \qquad (5)$$

As for $e^{(\mathbf{Q}\Delta)}$, we only need $I^{(\Delta)}$ through products $\frac{1}{\Delta}.W.I^{(\Delta)}$. Hence we compute these products iteratively to avoid the fill in. An analogous approach was used in (German 2001) for steady-state solution of Deterministic Stochastic Petri Nets (DSPN) but restricted to one deterministic event at any given time.

3. APPLICATION: THE $M/D/S/K$ QUEUE

In order to evaluate it, we apply our method to the M/D/S/K queue (Poisson arrival with rate λ, deterministic service time with duration D, S servers and finite capacity K). The $M/D/S/K$ queue is an interesting case study for several reasons. First, this model presents *multiple non exponential concurrent* activities (for $1 < S \leq K$). Second, no analytical results are available for the transient or even steady-state *distribution* of the queue length or other performance indices of this queueing model (for $S > 1$) unless $S = K$. It is thus significant to get quickly an approximation of these performance indices. Finally, due to the simplicity of the underlying process, we are able to easily compare our results with existing analytical or numerical tools for $S \leq 2$.

3.1 The approximate process

Let us recall that our approximation lies in the fact that non exponential events, i.e. the ends of (deterministic) service are recorded only at times $h\Delta$. For ease of notation, we set $\Delta = D/n$ so that n will be the indicator of the precision of the approximation in the rest of the paper.

S	K	$n=10$	$n=100$	$n=200$	$n=300$
1	10	101	911	1811	2711
2	10	597	46362	182712	409062
3	5	894	531984	4126459	13783434
3	10	2324	1416239	1416239	36736189

Table 1. Cardinalities of the state space of the approximate process

When we take into account these events, we may choose between two approximate strategies: either underestimate the (new) requested service time, choosing $n\Delta$ ("low" (L) choice) or in contrast overestimate this service time, choosing $(n+1)\Delta$ ("high" (H) choice). This leads to two series of parameters and approximate distributions that we denote by L or H superscript or $*$ for any of them; $n^{(*)}$ stands for n or $n+1$.

A state of the model is then $\langle c,x\rangle$ where

- c is the number of clients in the queue with $0 \leq c \leq K$;
- $x = (x_1, x_2, \ldots, x_{\min(c,S)})$ is the tuple of service states of each served client with $n^{(*)} \geq x_1 \geq x_2 \geq \ldots \geq x_{\min(c,S)}$.

The state $\langle 0,-\rangle$ corresponds to an empty queue and is the initial state of the system. States are ordered w.r.t. a kind of lexicographic order: the first state is $\langle 0,-\rangle$ and $\langle c,x\rangle < \langle c',x'\rangle$ iff $c < c'$ or $c = c'$ and $x = (x_1,\ldots,x_u) \leq x' = (x'_1,\ldots,x'_v)$ i.e. $u < v$ or $u = v$ and there is w such that $x_i = x'_i$ for all $i < w$ and $x_w < x'_w$ (lexical order between vectors).

There is no simple closed form for the cardinality of these state-spaces $S_n^{(L)}$ and $S_n^{(H)}$. They are obviously bounded by $(K+1)(n+1)^S$ but this is a very high bound. Table 1 gives the values of $S_n^{(H)}$ for some parameters.

In the $M/D/S/K$ case, the $\mathbf{Q}$ matrix defined in section 2 is very simple (in fact upper-triangular). From each state $\langle c,x\rangle$ (with $c < K$) the only exponential event is the Poisson arrival of a new client (rate λ), thus

$$\langle c',x'\rangle = \begin{cases} \langle c+1,x\rangle & (S \leq c < K) \\ \langle c+1,(n^{(*)}x_1,\ldots,x_{\min(c,S)})\rangle & (c < S) \end{cases}$$

If $c = K$ there is no transition from $\langle c,x\rangle$.

The $\mathbf{P}$ matrix records state changes due to non exponential events at times $h\Delta$. It has one only non null term per row (with $c > 0$) corresponding to decrease of remaining service times and possible ends of service followed by starts of service of waiting clients ($\mathbf{P}$ is lower-triangular). Let $e = |\{x_i \mid x_i = 1\}|$ be the number of servers which will end their service at next $h\Delta$ time in $s = \langle c,x\rangle$. Then $b = \max(\min(e, c-S), 0)$ is the number of servers that will start a new service at $h\Delta$ from s (such a server may have been busy or idle before $h\Delta$). Then, from $\langle c,x\rangle$, there is a transition with probability 1 to $\langle c',x'\rangle$ equal to

$$\langle c-e,(n^{(*)},\ldots,n^{(*)},x_1-1,\ldots,x_{\min(c,S)-e}-1)\rangle$$

with b occurrences of $n^{(*)}$ on the left side of x'. There is no transition from $\langle 0,-\rangle$ ($\mathbf{P}[1,1] = 0.0$).

3.2 Solving the approximate process

We compute approximate distributions (see section 2) for increasing values of n until they are as close as the required precision. Transient distributions are computed for discrete time horizons $h\Delta$. In all the paper, we use the L1 norm $\| \pi_2 - \pi_2 \| = \sum_i |\pi_1[i] - \pi_2[i]|$ to compare two probability distributions π_1 and π_2.

Let us denote $\pi_n^{(L)}(h)$ (resp. $\pi_n^{(H)}(h)$) the distribution at time $h\Delta$ computed for a given elementary time step $\Delta = D/n$ by taking approximate service delay $n\Delta$ (resp. $(n+1)\Delta$). We observed that each of the sequences $(\pi_n^{(L)}(h))_n$ and $(\pi_n^{(H)}(h))_n$ converges to $\pi^{(L)}(h)$ and $\pi^{(H)}(h)$ but that $\| \pi_n^{(L)}(h) - \pi_n^{(H)}(h) \|$ does not necessarily converge to 0. Moreover, several comparisons showed that depending on the parameters, one of the two sequences converges faster than the other and also that one of the two limits is closer to the exact distribution than the other, but we did not found any clear rule to choose between the two approximate distributions. These behaviours lead us to define the approximate distribution for n as a weighed sum of $\pi_n^{(L)}(h)$ and $\pi_n^{(H)}(h)$ based on their respective convergence rate:

$$\pi_n^{(a)}(h) = \frac{1}{d_n^{(L)} + d_n^{(H)}} \left(d_n^{(L)} \pi_n^{(L)}(h) + d_n^{(H)} \pi_n^{(H)}(h) \right)$$

with $d_n^{(*)} = \frac{1}{\|\pi_n^{(*)}(h) - \pi_{n'}^{(*)}(h)\|}$. and n' is the previous value used in the iterative approximation process. The steady-state approximate distribution is defined similarly.

For a given n, in both transient and steady-state cases, we first compute the matrices $\mathbf{P}_n^{(L)}$,$\mathbf{P}_n^{(H)}$, $\mathbf{Q}_n^{(L)}$,$\mathbf{Q}_n^{(H)}$. The initial probability vector is defined as $\pi_n(0)[1] = 1$ and $\pi_n(0)[j] = 0$ for all $j > 1$.

In the transient case (see (2)) we first compute the vector

$$\mathbf{V} = \frac{1}{\Delta} \pi_n(0) \left(e^{\mathbf{Q}_n^{(*)}\Delta} (\mathbf{P}_n^{(*)})^h \right)$$

then the product

$$\pi_n^{(*)}(h) = \mathbf{V}. \int_0^{\Delta} e^{\mathbf{Q}_n^{(*)}\tau} d\tau$$

Note that we never compute neither $e^{\mathbf{Q}_n^{(*)}\Delta}$ nor the integral matrices to avoid fill in.

In the steady-state case (see (3)), we first compute the solution $\pi_n^{(*,\Delta)}$ of $\pi = \pi.\mathbf{P}_n^{(*,\Delta)}$ with $\mathbf{P}_n^{(*,\Delta)} =$

ρ	time horizon 10D	50D	100D
0.1	40 - 10^{-4}	40 - 10^{-4}	40 - 10^{-4}
1.0	120 - 10^{-4}	140 - 10^{-3}	150 - 10^{-3}
10.0	110 - N/A	145 - N/A	150 - N/A

Table 2. Minimal n for $d \leq 10^{-3}$ and precision of the approximation for $S = 2, K = 5$

$e^{\mathbf{Q}_n^{(*)}\Delta}\mathbf{P}_n^{(*)}$ using an iterative method. Then we compute the product

$$\boldsymbol{\pi}_n^{(*)} = \frac{1}{\Delta}\boldsymbol{\pi}_n^{(*,\Delta)} \int_0^{\Delta} e^{\mathbf{Q}_n^{(*)}\tau} d\tau$$

From the two vectors $\boldsymbol{\pi}_n^{(L)}(h), \boldsymbol{\pi}_n^{(H)}(h)$ (or their steady-state versions) we derive $\boldsymbol{\pi}_n^{(a)}(h)$ and we iterate (increasing n by some step value) until $\| \boldsymbol{\pi}_n^{(a)}(h) - \boldsymbol{\pi}_{(n')}^{(a)}(h) \|$ reaches a given precision level ε.

3.3 Numerical results

We did several comparisons to validate our method and study its behaviour w.r.t. to the convergence and the precision of the approximation. All our computations were done on a Pentium-PC 2.6Ghz, 512MB, with Scilab 2.7.2 (INRIA 2002) under SuSE Linux. For all these measures, we have fixed $\varepsilon = 10^{-3}$, $K = 5, 10$. As we will show, the behaviour of the implementation depends heavily on the ratio $\rho = \lambda D/S$ which can be seen as the load indicator of the queue. Hence we have fixed $\lambda = 1.0$ in all experiments and we vary ρ in $\{0.1, 0.5, 1.0, 5.0, 10.0\}$, that is to say from a very low to a very high loaded system. For $S = 1$ we used several tools, mainly based on SPN: DSPNexpres (Lindemann 2003), GreatSPN (P.E. Group 2002), SPNica and TimeNET (German 2000). We also developed an exact solution based on the supplementary variable approach (Cox 1955*a*, German and Lindemann 1994) and we compared our approximation with this method too. For $S = 2$, we used DSPNepress-NG, the last version of DSPNexpress, the only analytical tool available to the best of our knowledge for two concurrent deterministic events (see (Lindemann 1998, Lindemann *et al.* 2000) for a detailed presentation). We do not report detailed results do to space limitation. We simply indicate the main trends we have observed.

3.3.1. Transient analysis In the transient case, we did computations for time horizons $t = 10D, 50D$ and $100D$ which correspond to $h = 10n, 50n$ and $100n$ for a given value of n (which varies during the computation of the approximation). To evaluate the convergence of the approximation, we computed the minimal n such that $d = \| \boldsymbol{\pi}_n^{(a)}(h) - \boldsymbol{\pi}_{(n-1)}^{(a)}(h) \| \leq 10^{-3}$. Results for $S = 2$ are given in Table 2. In all cases, the minimal n was between 100 and 200 and is increasing slowly with t. For $K = 10$, we have observed however that for $\rho \gg 1$, the variation on $\boldsymbol{\pi}_n^{(a)}(h)$ first decreases ($n \leq 300$) but then increased with n. This indicates clearly that for large n, numerical computations become unstable. In all cases, we observed a difference lower than 10^{-3} between our approximation and the result given by DSPNexpress-NG.

3.3.2. Steady-state analysis We did the same experiments for the steady-state case. As suspected by the transient analysis, the required n to reach a fixed precision depends largely on ρ. In fact, since we force the end of service time at $n\Delta$ only, our method provides an almost exact result when either the servers are almost always idle ($\rho \ll 1$) or busy ($\rho \gg 1$). We recognize this behaviour both in the higher values of n and the lower precision. This property of our method is also illustrated by the results (not reported here due to lack of space) for the special case $S = K$ for which the exact result is known to be equal to the $M/M/S/S$ queue. In this infinite server case, every end of service is immediately followed by a beginning of service when the system is heavily loaded. In these cases we have obtained small values of n to get the approximation. In contrast, for $\rho \approx 1$, the convergence is slow ($n \approx 200 - 300$) and the approximation is not so good (precision=10^{-3}). For $K = 5$ (resp. $K = 10$) and $\rho = 4.5$ (resp. $\rho > 1.75$), we have noticed that DSPNexpress-NG gives erroneous probabilities (i.e. negative values) so that we could not compare our result with an "exact" numerical value.

4. CONCLUSIONS AND PERSPECTIVES

We have presented a new approximate method for stochastic processes with multiple non Markovian concurrent activities. Contrary to the other methods, we have approximated the model and applied an exact method rather than the opposite way to do. The key factor for the quality of this approximation is that the occurrence of Markovian events are not approximated as it would be in a naive discretization process. Furthermore, the design of its analysis is based on robust numerical methods (i.e. uniformization) and the steady-state and transient cases are handled similarly.

The examples have shown that our method is robust. For instance when analysing an heavy queue the precision is still correct (i.e. 10^{-3} whereas DSPNexpress-NG (a very efficient tool in general) gives negative probabilities!

We are currently extending our method in two directions. The first one is to adapt the approach to models exhibiting very different durations (stiff models). In such cases, the determination of Δ uniquely based on the smallest deterministic delay would lead to intractable matrices and vectors.

The second extension is related to the introduction of tensorial expressions for the matrices in order to manage the space complexity associated to systems with many non Markovian concurrent activities.

Acknowledgments We would like to thank the authors of the different solvers we have used for providing us the last version of their software.

REFERENCES

Ajmone Marsan, M. and G. Chiola (1987). On Petri nets with deterministic and exponentially distributed firing times. In: *Advances in Petri Nets 1987* (G. Rozenberg, Ed.). pp. 132–145. Number 266 In: *LNCS*. Springer–Verlag.

Cox, D. R. (1955*a*). The analysis of non-Markov stochastic processes by the inclusion of supplementary variables. *Proc. Cambridge Philosophical Society (Math. and Phys. Sciences)* **51**, 433–441.

Cox, D. R. (1955*b*). A use of complex probabilities in the theory of stochastic processes. *Proc. Cambridge Philosophical Society* pp. 313–319.

German, R. (2000). *Performance analysis of communication systems*. John Wiley & Sons,Ltd. Chichester, England.

German, R. (2001). Iterative analysis of Markov regenerative models. *Performance Evaluation* **44**, 51–72.

German, R. and C. Lindemann (1994). Analysis of stochastic Petri nets by the method of supplementary variables. *Performance Evaluation* **20**(1–3), 317–335. special issue: Peformance'93.

German, R., D. Logothesis and K.S. Trivedi (1995). Tran-
sient analysis of Markov regenerative stochastic Petri nets: A comparison of approaches. In: *Proc. of the 6th International Workshop on Petri Nets and Performance Models*. IEEE Computer Society Press. Durham, NC, USA. pp. 103–112.

Gross, D. and D.R. Miller (1984). The randomization technique as a modeling tool an solution procedure for transient markov processes. *Operations Research* **32**(2), 343–361.

INRIA (2002). Scilab home page: `http://www.scilab.org`.

Lindemann, C. (1998). *Performance Modelling with Deterministic and Stochastic Petri Nets*. John Wiley & Sons.

Lindemann, C. (2003). DSPNexpress home page: `http://www.dspnexpress.de.`.

Lindemann, C., a. Klemm, M. Lohmann and O. Walhorst (2000). Quantitative system evaluation with DSPNexpress 2000. In: *Proc. of the* 2^{nd} *Int. Workshop on Software and Performance (WOSP)*. Ottawa, canada. pp. 12–17.

Lindemann, C., A. Reuys and A. Thümmler (1999). DSPNexpress 2.000 performance and dependability modeling environment. In: *Proc. of the* 29^{th} *Int. Symp. on Fault Tolerant Computing*. Madison, Wisconsin.

Lindemann, C. and G.S. Schedler (1996). Numerical analysis of deterministic and stochastic Petri nets with concurrent deterministic transitions. *Performance Evaluation* **27–28**, 576–582. special issue: Proc. of PERFORMANCE'96.

P.E. Group (2002). GreatSPN home page: `http://www.di.unito.it/~greatspn`.

Stewart, W. J. (1994). *Introduction to the numerical solution of Markov chains*. Princeton University Press, USA.