Discrete-time Markovian stochastic Petri nets

Gianfranco Ciardo

Department of Computer Science
College of William and Mary
Williamsburg, VA 23187-8795
ciardo@cs.wm.edu

Abstract

We revisit and extend the original definition of discrete-time stochastic Petri nets, by allowing the firing times to have a "defective discrete phase distribution". We show that this formalism still corresponds to an underlying discrete-time Markov chain. The structure of the state for this process describes both the marking of the Petri net and the phase of the firing time for of each transition, resulting in a large state space. We then modify the well-known power method to perform a transient analysis even when the state space is infinite, subject to the condition that only a finite number of states can be reached in a finite amount of time. Since the memory requirements might still be excessive, we suggest a bounding technique based on truncation.

[1] This research was partially supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-19480 while in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23681-0001.

1 Introduction

In the past decade, stochastic Petri nets (SPNs) have received much attention from researchers in the performance and reliability arena and have been extensively applied to the performance and reliability modeling of computer, communication, manufacturing, and aerospace systems [4, 5, 7, 10, 23]. While there is agreement on the appropriateness of SPNs as a description formalism for a large class of systems, two radically different solution approaches are commonly employed: simulation and state-space-based analysis. Simulation allows to associate general distributions to the duration of activities (SPN transitions), but it requires multiple runs to obtain meaningful statistics. This problem is particularly acute in reliability studies, where many runs might be required to obtain tight confidence intervals. With simulation, the state of the SPN is composed of the *marking*, describing the structural state of the SPN, and the *remaining firing times*, describing how long each transition in the SPN must still remain enabled before it can fire. The simulated time θ is advanced by *firing* the transition with the smallest remaining firing time.

State-space-based analysis has been mostly applied to SPNs whose underlying process is a continuous-time Markov chain (CTMC), that is, to SPNs with exponentially distributed firing times [3, 12, 25, 26]. Except for numerical truncation and roundoff, exact results are obtained, but the approach has two limitations: the number of SPN markings increases combinatorially, rendering unfeasible the solution of large models, and generally-distributed activities must be modeled using "phase-type (PH) expansion" [15]. PH distributions can approximate any distribution arbitrarily well, but it is difficult to exploit this fact in practice because the expansion exacerbates the state-space size problem.

Discrete distributions for the timing of SPNs have received less attention. This is unfortunate, since deterministic distributions (constants) are often needed to model low-level phenomena in both hardware and software, and the geometric distribution is the discrete equivalent of the exponential distribution and can approximate it arbitrarily well as the size of the step decreases. Furthermore, there is evidence supporting the use of deterministic instead of exponential distributions when modeling parallel programs [1].

If all the firing distributions are geometric with the same step, the underlying process is a discrete-time Markov chain (DTMC) [25]. Such SPNs can model synchronous behavior, as well as the main aspect of asynchronous systems: the uncertainty about the ordering of quasi-simultaneous events. A DTMC is described by a square one-step state transition probability matrix Π and an initial state probability vector $\pi^{[0]}$. The state probability vector at step n can be obtained with the iteration (*power method*): $\pi^{[n]} = \pi^{[n-1]}\Pi$. This result was extended in [11] to include immediate transitions, which fire in zero time, and geometric firing distributions with steps multiple of a basic unit step, possibly with parameter equal one, that is, constants. [29] restates these results in more detail, and uses the concept of weight to break the ties, following [3] and, more closely, [13]. Generalized Timed Petri Nets (GTPN) have also been proposed [19], where the steps of the geometric firing times for each transition can be arbitrary, unrelated, real numbers. A DTMC can be obtained by embedding, but the analysis is restricted to steady-state behavior and the state space of the DTMC can be infinite even when the underlying untimed PN has a finite reachability set. Analogous considerations hold for D-timed PNs [30].

We generalize and formalize the results in [13] and show how, using phase-expansion, a DTMC can be obtained even if the firing time distributions are not geometric, as long as firings can occur only at some multiple of a unit step. The state can then be described by the marking plus the phase of each transition. This extends the class of SPNs that can be solved analytically, but two limitations still exist: the existence of a basic step and the size of the state space. By using a fine step, arbitrary steps can be approximated, but this increases the state space.

Approaches to solve models with a large state space have been proposed for both steady-state and transient analysis. [6] considers the reliability study of a SPN with exponentially distributed firing times, under the condition that the reachability graph is acyclic. The underlying CTMC is then acyclic as well, and a state can be discareded as soon as the transitions out of it have been explored, resulting in an algorithm offering large savings in memory and computations with respect to traditional numerical approaches. However, acyclic state spaces arise only in special cases, such as reliability models of non-repairable systems.

For transient analysis of a general CTMC, Jensen's method [21], also called uniformization [17, 27], is widely adopted. [18] outlines a dynamic implementation of the algorithm, where the state space is explored as the computation of the transient probability vector proceeds, not in advance, as normally done. This allows to obtain a transient solution even if the state space is infinite, provided that the transition rates have an upper bound.

If the CTMC contains widely different rates, the number of matrix-vector multiplications required by uniformization can be excessive. Proposals to alleviate this problem are selective randomization [24] and adaptive uniformization [28], both based on the idea of allowing different uniformization rates, according to the set of states that can be reached at each step. The latter, in addition, can be used with infinite state spaces even if the rates have no upper bound. However, the method can incur a substantial overhead, and it appears that an adaptive step is advantageous only in special cases or for short time horizons.

In Sections 2, 3, and 4 we define the underlying untimed PN model, the class of DDP distributions used for the temporization of a PN, and the resulting DDP-SPN formalism, respectively. Section 5 discusses the numerical solution of a DDP-SPN, by building and solving its underlying stochastic process, a DTMC. Section 6, examines approaches to cope with large state spaces.

2 The PN formalism

We recall the (extended) PN formalism used in [12, 14]. A PN is a tuple $\left(P, T, D^{-}, D^{+}, D^{\circ}, \succ, g, \mu^{[0]}\right)$ where:

- P is a finite set of places, which can contain tokens. A marking $\mu \in \mathbb{N}^{|P|}$ defines the number of tokens in each place $p \in P$, indicated by μ_p (when relevant, a marking should be considered a column vector). D^{-}, D^{+}, D°, and g are "marking-dependent", that is, they are specified as functions of the marking.

- T is a finite set of transitions. $P \cap T = \emptyset$.

- $\forall p \in P, \forall t \in T, \forall \mu \in \mathbb{N}^{|P|}$, $D^{-}_{p,t}(\mu) \in \mathbb{N}$, $D^{+}_{p,t}(\mu) \in \mathbb{N}$, and $D^{\circ}_{p,t}(\mu) \in \mathbb{N}$ are the multiplicities of the input arc from p to t, the output arc from t to p, and the inhibitor arc from p to t, when the marking is μ, respectively.

- $\succ \subseteq T \times T$ is an acyclic (pre-selection) priority relation.

- $\forall t \in T, \forall \mu \in \mathbb{N}^{|P|}, g_t(\mu) \in \{0, 1\}$ is the guard for t in marking μ.

- $\mu^{[0]} \in \mathbb{N}^{|P|}$ is the initial marking.

Places and transitions are drawn as circles and rectangles, respectively. The number of tokens in a place is written inside the place itself (default is zero). Input and output arcs have an arrowhead on their destination, inhibitor arcs have a small circle. The multiplicity is written on the arc (default is the constant 1); a missing arc indicates that the multiplicity is the constant 0. The default value for guards is the constant 1.

A transition $t \in T$ is enabled in marking μ iff all the following conditions hold:

1. $g_t(\mu) = 1$.

2. $\forall p \in P, D^{-}_{p,t}(\mu) \leq \mu_p$.

3. $\forall p \in P, D^{\circ}_{p,t}(\mu) > \mu_p$ or $D^{\circ}_{p,t}(\mu) = 0$.

4. $\forall u \in T, u \not\succ t$ or u is not enabled in μ (well defined because $\succ$ is acyclic).

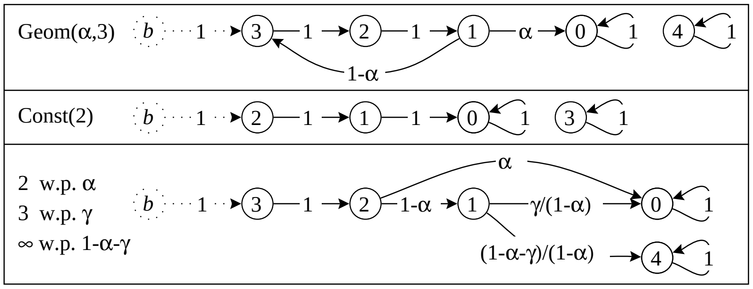

Figure 1: Examples of DDP distributions.

Let $\mathcal{E}(\mu)$ be the set of transitions enabled in marking μ. A transition $t \in \mathcal{E}(\mu)$ can fire, causing a change to marking $\mathcal{M}(t, \mu)$, obtained from μ by subtracting the "input bag" $D^{-}_{\bullet,t}(\mu)$ and adding the "output bag" $D^{+}_{\bullet,t}(\mu)$ to it: $\mathcal{M}(t, \mu) = \mu - D^{-}_{\bullet,t}(\mu) + D^{+}_{\bullet,t}(\mu) = \mu + D_{\bullet,t}(\mu)$, where $D = D^{+} - D^{-}$ is the incidence matrix. $\mathcal{M}$ can be extended to its reflexive and transitive closure by considering the marking reached from μ after firing a sequence of transitions. The *reachability set* is given by $\mathcal{R} = \{\mu : \exists \sigma \in T^{*} \wedge \mu = \mathcal{M}(\sigma, \mu^{[0]})\}$, where T^{*} indicates the set of transition sequences.

3 Discrete phase distributions

We now define the class $\mathcal{D}$ of (possibly defective) discrete phase (DDP) distributions, which will be used to specify the duration of a firing time in a SPN. A random variable X is said to have a DDP distribution, $X \sim \mathcal{D}$, iff there exists an absorbing DTMC $\{A^{[k]} : k \in \mathbb{N}\}$ with finite state space $\mathcal{A} = \{0, 1, \ldots, n\}$ and initial probability distribution given by $[\Pr\{A^{[0]} = i\}, i \in \mathcal{A}]$, such that states $\mathcal{A} \setminus \{0, n\}$ are transient and states $\{0, n\}$ are absorbing, and X is the time to reach state 0: $X = \min\{k \geq 0 : A^{[k]} = 0\}$. If $\Pr\{A^{[0]} = 0\} > 0$, the distribution has a mass at the origin. If $\Pr\{A^{[0]} = i\} > 0$ and state i can reach state n, the distribution is (strictly) defective.

$\mathcal{D}$ is the smallest class containing the distributions Const(0), Const(1), and Const(∞) and closed under:

- Finite convolution: if $X_1 \sim \mathcal{D}$ and $X_2 \sim \mathcal{D}$, then $X = X_1 + X_2 \sim \mathcal{D}$.

- Finite weighted sum: if $X_1 \sim \mathcal{D}$, $X_2 \sim \mathcal{D}$ and $B \in \{0, 1\}$ is a Bernoulli random variable, then $X = BX_1 + (1 - B)X_2 \sim \mathcal{D}$.

- Infinite geometric sum: if $\{X_k \sim \mathcal{D} : k \in \mathbb{N}^{+}\}$ is a family of iid's and N is a geometric random variable, then $X = \sum_{1 \leq k \leq N} X_k \sim \mathcal{D}$.

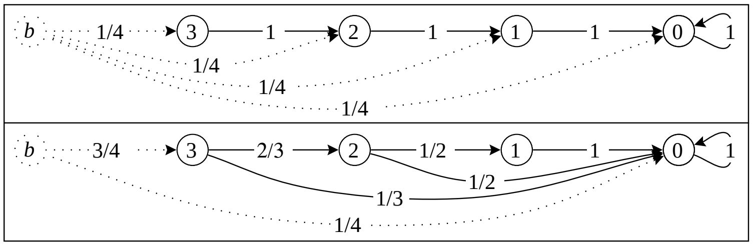

Figure 2: Equivalent DTMC representations

The geometric and modified geometric distributions with arbitrary positive integer step, $\mathrm{Geom}(\alpha, \omega)$ and $\mathrm{ModGeom}(\alpha, \omega)$, $0 \leq \alpha \leq 1$, $\omega \in \mathbb{N}^+$, the constant non-negative integer distribution, $\mathrm{Const}(\omega)$, $\omega \in \mathbb{N}$, and any discrete distribution with finite non-negative integer support are special cases of DDP distributions. An example of a random variable with non-negative integer support which does not have a DDP distribution is N^2, where $N \sim \mathrm{Geom}(\alpha)$.

Fig. 1 shows examples of DDP distributions. The "initial state" b, for begin, has zero sojourn time and is introduced to represent graphically the initial probability distribution. We use this representation since it allows to estimate the "complexity" of a DTMC by counting the number of nodes and arcs in its graph. For simplicity, the last state, e.g., 4 for $\mathrm{Geom}(\alpha, 3)$ and 3 for $\mathrm{Const}(2)$, can be omitted if it is not reachable from b (if the distribution is actually not defective). Unfortunately, the DTMC corresponding to a given DDP distribution might not be unique, even if the number of states if fixed. For example, the time X to reach state 0 for the DTMCs in Fig. 2, both with five nodes and seven arcs, has distribution $\mathrm{Unif}(0, 3)$, that is, $\Pr\{X = i\} = 1/4$, for $i \in \{0, 1, 2, 3\}$.

4 The DDP-SPN formalism

SPNs are obtained when the time that must elapse between the instant a transition becomes enabled and the instant it can fire, or firing time, is a random variable. By restricting the firing times distributions to $\mathcal{D}$, we obtain the DDP-SPNs, corresponding to a stochastic process where the state has the form $s = (\mu, \phi) \in \mathbb{N}^{|P|} \times \mathbb{N}^{|T|}$. The structural component μ is simply the current marking. The timing component ϕ describes the current "phases", the state for the DTMC chosen to encode the DDP distribution associated with the firing time of each transition. The firing time of a transition t elapses when its phase ϕ_t reaches 0. Formally, a DDP-SPN is a tuple $\left(P, T, D^-, D^+, D^\circ, \succ, g, \mu^{[0]}, \Phi, G, F, \phi^{[0]}, \not\succ, w\right)$ where:

- $\left(P, T, D^-, D^+, D^\circ, \succ, g, \mu^{[0]}\right)$ define a PN.
- $\forall t \in T, \forall \mu \in \mathcal{R}, \Phi_t(\mu) \subset \mathbb{N}$ is the finite set of possible phases in which transition t can be when the marking is μ.

- $\forall \mu \in \mathcal{R}, \forall t \in \mathcal{E}(\mu), \forall i,j \in \Phi_t(\mu), G_t(\mu,i,j)$ is the probability that the phase of t changes from i to j at the end of one step, when t is enabled in marking μ. Hence, $\sum_{j \in \Phi_t(\mu)} G_t(\mu,i,j) = 1$.
- $\forall \mu \in \mathcal{R}, \forall u \in \mathcal{E}(\mu), \forall t \in T, \forall i \in \Phi_t(\mu), \forall j \in \Phi_t(\mathcal{M}(u,\mu)), F_{u,t}(\mu,i,j)$ is the probability that the phase of t changes from i to j when u fires in marking μ. Hence, $\sum_{j \in \Phi_t(\mathcal{M}(u,\mu))} F_{u,t}(\mu,i,j) = 1$.
- $\forall t \in T, \phi_t^{[0]} \in \Phi_t(\mu^{[0]})$ is the phase of t at time 0.
- $\succ\!\!\succ \subseteq T \times T$ is an acyclic (post-selection) priority relation.
- $\forall \mu \in \mathcal{R}, \forall S \subseteq \mathcal{E}(\mu), \forall t \in S, w_{t|S}(\mu) \in \mathbb{R}^+$ is the firing weight for t when S is the set of candidates to fire in marking μ.

A transition $t \in T$ is said to be a *candidate* (to fire) in state $s = (\mu, \phi)$ iff all the following conditions hold:

1. $t \in \mathcal{E}(\mu)$.
2. $\phi_t = 0$.
3. $\forall u \in T, u \not\succ\!\!\succ t$ or u is not a candidate in s (remember that $\succ\!\!\succ$ is acyclic).

Let $\mathcal{C}(s)$ be the set of candidates in state s. $G_t(\mu, \bullet, \bullet)$ is the one-step transition probability matrix of the DTMC $\{\phi_t^{[k]} : k \in \mathbb{N}\}$, with state space $\Phi_t(\mu)$, corresponding to the DDP-distributed firing time for transition t in marking μ in isolation, that is, assuming that no other transition firing affects the firing time of t. However, if another transition u fires before t, leading to marking μ', the phase ϕ_t of t will change according to the distribution $F_{u,t}(\mu, \phi_t, \bullet)$. Furthermore, after the firing of u, the phase of t will evolve according to $G_t(\mu', \bullet, \bullet)$, which might differ from $G_t(\mu, \bullet, \bullet)$, it can even have a different state space, $\Phi_t(\mu')$ instead of $\Phi_t(\mu)$.

We stress that pre-selection and post-selection have a different semantic. Only in the case of immediate transitions the two become equivalent. Assume that only t and u satisfy the input, inhibitor, and guard conditions in μ. We have three options, resulting in three different behaviors:

- Specify a pre-selection priority between them, for example $t \succ u$, so that u will not be enabled when t is. This means that the phase ϕ_t of t evolves according to $G_t(\mu, \bullet, \bullet)$, while ϕ_u does not. The same effect would be achieved using a guard $g_u(\mu) = 0$.

- Specify no pre-selection priority, but a post-selection priority between them, for example $t \succ u$. This means that the phases of both t and u evolve in μ. The first one to reach phase 0 will fire but, in case of a tie, t will be chosen. However, if $\phi_u = 0$ when t fires and if $F_{t,u}(\mu, 0, 0) = 1$, u might be a candidate in the new marking, and fire immediately after t.

- Specify neither a pre-selection nor a post-selection priority between them. Then, as in the previous case, t and u are in a race to reach phase 0, but a tie is now resolved by a probabilistic choice according to the the weights: $\hat{w}_{t|\{t,u\}}(\mu)$ and $\hat{w}_{u|\{t,u\}}(\mu)$, respectively, where $\hat{w}$ is a normalization of w to ensure that the weights of the candidates in a marking sum to one.

Let $(\mu^{[n]}, \phi^{[n]})$ be the state of the DDP-SPN at step n. Then, the process $\{(\mu^{[n]}, \phi^{[n]}) : n \in \mathbb{N}\}$ is a DTMC with state space $\mathcal{S} \subseteq \mathbb{N}^{|P|} \times \mathbb{N}^{|T|}$. Its one-step transition probability matrix Π is determined by considering the possibility of simultaneous firings. Consider a state $s = (\mu, \phi)$. If $\mathcal{C}(s) \neq \emptyset$, one of the candidates will fire immediately, and the sojourn time in s is zero. Otherwise, the sojourn time in s is one. Following GSPN [3] terminology, we call s a vanishing or tangible state, respectively. Hence, s is tangible iff $\phi > 0$.

Let $S_{s,s'}$ be the set of possible event sequences events leading from a tangible state $s = (\mu, \phi)$ to a tangible state $s' = (\mu', \phi')$ in one time step:

$$\begin{aligned} S_{s,s'} = \Big\{ & \sigma = (\mu^{(0)}, \phi^{(0)}, t^{(0)}, \mu^{(1)}, \phi^{(1)}, t^{(1)}, \ldots \mu^{(n-1)}, \phi^{(n-1)} t^{(n-1)}, \mu^{(n)}, \phi^{(n)}) : \\ & n \geq 0, \mu^{(0)} = \mu, \mu^{(n)} = \mu', \phi^{(n)} = \phi', \\ & \forall t \in \mathcal{E}(\mu), G_t(\mu, \phi_t, \phi_t^{(0)}) > 0, && (1) \\ & \forall i, 0 \leq i < n, t^{(i)} \in \mathcal{C}(\mu^{(i)}, \phi^{(i)}), \mu^{(i+1)} = \mathcal{M}(t^{(i)}, \mu^{(i)}), && (2) \\ & \forall t \in T, F_{t^{(i)},t}(\mu^{(i)}, \phi_t^{(i)}, \phi_t^{(i+1)}) > 0 \Big\}. && (3) \end{aligned}$$

(1) considers the one-step evolution of the phases for the enabled transitions in isolation, while (2) and (3) consider the sequentialized firing in zero time of zero or more transitions at the end of the one-step period. Hence, $(\mu^{(i)}, \phi^{(i)})$ is a vanishing state, for $0 \leq i < n$.

The value of the nonzero entries of Π is obtained by summing the probability of all possible sequences leading from s to s':

$$\begin{aligned} \Pi_{s,s'} = \sum_{\sigma \in S_{s,s'}} & \left(\prod_{t \in \mathcal{E}(\mu)} G_t(\mu, \phi_t, \phi_t^{(0)}) \right) \\ & \cdot \left(\prod_{i=0}^{n-1} \hat{w}_{t^{(i)}|\mathcal{C}(\mu^{(i)}, \phi^{(i)})}(\mu^{(i)}) \left(\prod_{t \in T} F_{t^{(i)},t}(\mu^{(i)}, \phi_t^{(i)}, \phi_t^{(i+1)}) \right) \right) \end{aligned}$$

In a practical implementation, Π is computed one row at a time. The complexity of computing row s of Π can be substantial, depending on the length and number of sequences

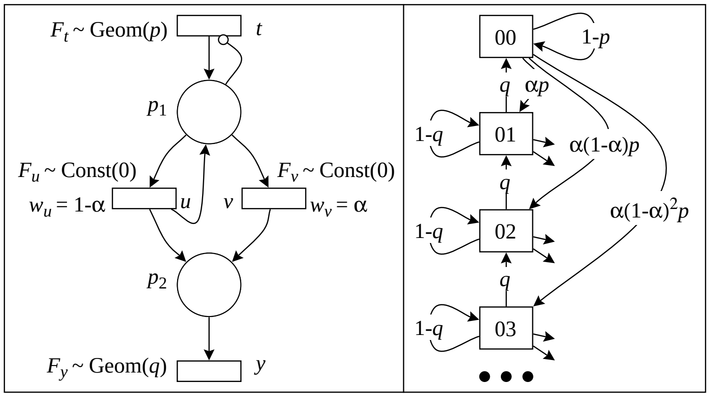

Figure 3: $(0,0)$ can reach an infinite number of markings in one time step.

in $\bigcup_{s'} S_{s,s'}$. If $\bigcup_{s'} S_{s,s'}$ is infinite, special actions must be taken. This can happen for two reasons:

- $\mathcal{R}$ is itself infinite, and state s can reach an infinite number of states in a single step. Consider, for example, a single queue with batch arrivals of size $N > 0$, where $N \sim \text{Geom}(\alpha)$, as in Fig. 3. Following the firing of t, a geometrically distributed number of tokens will be placed in p_2: when the token is finally removed from p_1 (by the firing of v), p_2 contains N tokens with probability $\alpha(1-\alpha)^{N-1}$. This represents a batch arrival of size N at the server modeled by place p_2 and transition y. Unfortunately, finiteness of $\mathcal{R}$ is an undecidable question for the class of Petri nets we defined, since transition priorities alone make them Turing equivalent [2].

- $S_{s,s'}$ can be infinite for a particular s'. If $\mathcal{R}$ is finite, this requires the presence of arbitrarily long paths over a finite set of vanishing states, just as for a "vanishing loop" in a GSPN [11]. In a practical implementation, these cycles can be detected and managed appropriately.

The size of the DTMC underlying a DDP-SPN is affected by the choice of the representation for the DDP distributions involved. Consider, for example, the DDP-SPN in Fig. 4(a), and assume that transitions t_1, t_2, and t_3 have firing time distributions Const(1), Unif$(0,3)$, and Const(2), respectively. The corresponding DTMCs obtained using the two representations of Fig. 2 for Unif$(0,3)$ are shown in Fig. 4(b) and 4(c), respectively. The number of states is ten in the first case, seven in the second (the value of ϕ_t is specified as "$\bullet$" whenever t is not enabled and either it cannot become enabled again or its phase is going to be reset upon becoming enabled). The difference between the size of the two DTMCs is due to a *lumping* [22] of the states, and it would be even greater if t_3 had a more complex distribution. By postponing the probabilistic decision as much as possible, the second DTMC lumps

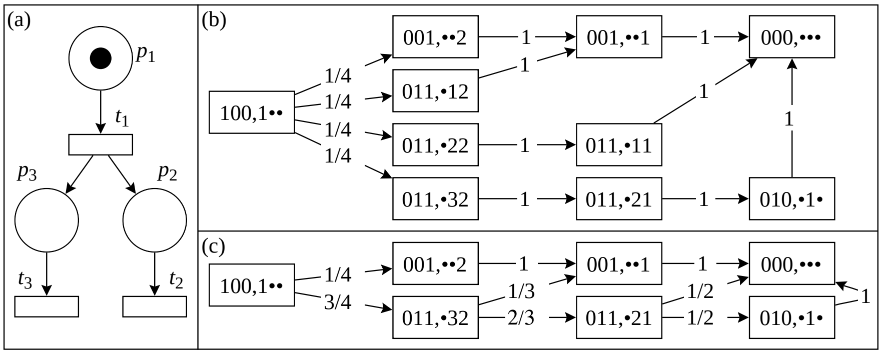

Figure 4: The effect of equivalent $\mathrm{Unif}(0, 3)$ representations.

states $(011, \bullet 12)$, $(011, \bullet 22)$, and $(011, \bullet 32)$ of the first DTMC into a single one, $(011, \bullet 32)$, and states $(011, \bullet 11)$ and $(011, \bullet 21)$ into $(011, \bullet 21)$.

5 Analysis of DDP-SPNs

When using a SPN to model a system, a reward rate ρ_μ is associated to each marking μ. Starting from $\{(\mu^{[n]}, \phi^{[n]}) : n \in \mathbb{N}\}$, it is then possible to define two continuous-parameter processes: $\{y(\theta), \theta \geq 0\}$, describing the instantaneous reward rate at time θ: $y(\theta) = \rho_{\mu(\theta)}$, where $\mu(\theta) = \mu^{[\max\{n \leq \theta\}]}$, and $\{Y(\theta), \theta \geq 0\}$, describing the reward accumulated up to time θ, $Y(\theta) = \int_0^\theta \rho_{\mu(\tau)} d\tau$.

We consider the computation of the expected value of $y(\theta_F)$ and $Y(\theta_F)$ for finite values of θ_F. Let $\pi^{[n]} = \left[\pi_s^{[n]}\right] = \left[\Pr\{s^{[n]} = s\}\right]$ be the state probability vector at time n. Once the state-space $\mathcal{S}$ corresponding to the initial state $(\mu^{[0]}, \phi^{[0]})$ has been generated, any initial probability vector over $\mathcal{S}$ can be used for the initial probability vector $\pi^{[0]}$, there is no requirement to use a vector having a one in position $(\mu^{[0]}, \phi^{[0]})$ and a zero elsewhere. From $\pi^{[0]}$, we can obtain $\pi^{[n]}$ iteratively, performing n matrix-vector multiplications:

$$\pi^{[n]} = \pi^{[n-1]}\Pi \tag{4}$$

Since the DTMC can change state only at integer times, $\pi(\theta) = \pi^{[n]}$ for $\theta \in [n, n+1)$. Practical implementations assume that the state space is finite and that the transition probability matrix Π is computed before starting the iterations. The following shows the pseudo-code to compute $E[y(\theta_F)]$ and $E[Y(\theta_F)]$ with the "power method":

```
1. "compute $\mathcal{S}$, $\Pi$, and $\pi^{[0]}$";
2. $Y \leftarrow 0$; $\pi \leftarrow \pi^{[0]}$;
3. for $n = 1$ to $\lfloor \theta_F \rfloor$ do
```

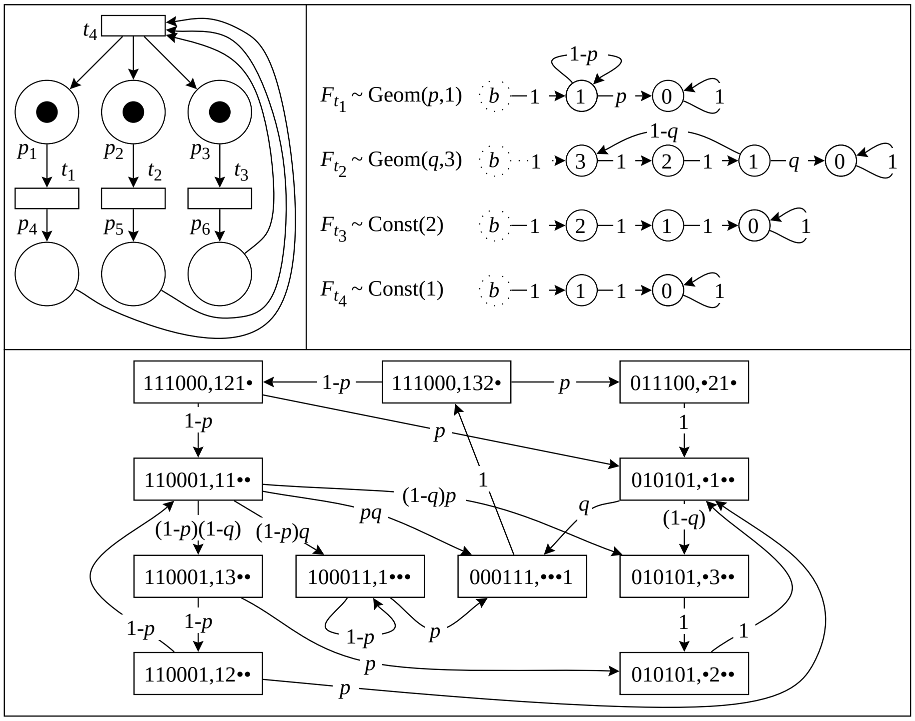

Figure 5: A DDP-SPN with an ergodic underlying DTMC.

```
4.        $Y = Y + \sum_{(\mu,\phi)\in\mathcal{S}} \rho_\mu \pi_{(\mu,\phi)}$;
5.        $\pi \leftarrow \pi\Pi$;
6.  $E[Y(\theta_F)] \leftarrow Y + (\theta_F - \lfloor\theta_F\rfloor) \sum_{(\mu,\phi)\in\mathcal{S}} \rho_\mu \pi_{(\mu,\phi)}$;
7.  $E[y(\theta_F)] \leftarrow \sum_{(\mu,\phi)\in\mathcal{S}} \rho_\mu \pi_{(\mu,\phi)}$;
```

If the state space $\mathcal{S}$ is finite, it is possible to approximate the steady-state probability vector $\pi^* = \lim_{n\to\infty} \pi^{[n]}$ by iterating the power method long enough. If the DTMC is ergodic, though, other numerical approaches are preferable, based on the relation $\pi^* = \pi^*\Pi$, which can be rewritten as the homogeneous linear system $\pi^*(\Pi - I) = 0$, subject to $\sum_{s\in\mathcal{S}} \pi^*_s = 1$. Fast iterative methods such as successive over-relaxation (SOR) [12] or multilevel methods [20] can then be employed, although their convergence is not guaranteed. Fig. 5 offers an example of an ergodic DTMC obtained from a DDP-SPN.

6 Coping with large state spaces

The power method algorithm described requires to generate the state space $\mathcal{S}$ and Π, and then to iterate using Eq. (4), hence it assumes a finite $\mathcal{S}$. However, a "dynamic" state space exploration has been proposed to remove this restriction [16, 18]. The general idea is to start from the initial state, or set of initial states and iteratively compute both the set of reachable states and the probability of being in them after n steps, for increasing values of n. The approach has been proposed for the transient analysis of CTMCs using uniformization [17, 21, 27], where, in practice, the iterations must be stopped at a large but finite n, thus resulting in a truncation error which can be bounded. However, the same approach is even more appropriate for the transient analysis of DTMCs, since, in this case, no truncation is required: the exact number of steps to be considered is determined by the time θ_F at which the results are desired.

Let $\mathcal{S}^{[n]}$ be the set of states explored at step n. States in $\mathcal{S} \setminus \mathcal{S}^{[n]}$ have zero probability at step n, given the initial state(s). Then, $\mathcal{S}^{[0]}$ is completely determined by $\pi^{[0]}$, which is given, and $\mathcal{S}^{[n]}$ is obtained from $\mathcal{S}^{[n-1]}$ by considering the nonzero entries in $\Pi_{s,\bullet}$ for each $s \in \mathcal{S}^{[n-1]}$. The pseudo-code for this modified power method algorithm is:

```
$\Pi \leftarrow 0$; $\pi \leftarrow 0$; $\theta \leftarrow 0$; $\mathcal{S} \leftarrow \{(\mu^{[0]}, \phi^{[0]})\}$; $\mathcal{N} \leftarrow \mathcal{S}$; $\pi_{(\mu^{[0]},\phi^{[0]})} \leftarrow 1.0$;
for $n = 1$ to $\lfloor \theta_F \rfloor$ do
    $Y = Y + \sum_{(\mu,\phi)\in\mathcal{S}} \rho_\mu \pi_{(\mu,\phi)}$;
    $\mathcal{N}' \leftarrow \emptyset$;
    while $\exists s \in \mathcal{N}$ do
        for each $s'$ such that $S_{s,s'} \neq \emptyset$ do
            "compute $\Pi_{s,s'}$";
            if $s' \notin \mathcal{S}$ then
                $\mathcal{N}' \leftarrow \mathcal{N}' \cup \{s'\}$; $\mathcal{S} \leftarrow \mathcal{S} \cup \{s'\}$;
        $\mathcal{N} \leftarrow \mathcal{N} \setminus \{s\}$;
    $\mathcal{N} \leftarrow \mathcal{N}'$;
    $\pi \leftarrow \pi\Pi$;
$E[Y(\theta_F)] \leftarrow Y + \sum_{(\mu,\phi)\in\mathcal{S}} \rho_\mu \pi_{(\mu,\phi)}(\theta_F - \lfloor \theta_F \rfloor)$;
$E[y(\theta_F)] \leftarrow \sum_{(\mu,\phi)\in\mathcal{S}} \rho_\mu \pi_{(\mu,\phi)}$;
```

At the beginning of the n-th iteration, $\mathcal{S}$ and $\mathcal{S} \setminus \mathcal{N}$ contain the states reachable in less than n and $n-1$ steps, respectively. The rows $\Pi_{s,\bullet}$ for the states $s \in \mathcal{S} \setminus \mathcal{N}$ have been built in previous iterations, while those corresponding to states $s \in \mathcal{N}$ still need to be computed. During the n-th iteration, $\mathcal{N}'$ accumulates the states reachable in exactly n, but not fewer, steps. These states will be explored at the next iteration. This algorithm allows to study a DDP-SPN regardless of whether $\mathcal{S}$ is finite or not, provided that:

- θ_F is finite (transient analysis).
- A finite set of states has nonzero initial probability: $|\{s : \pi_s^{[0]} > 0\}| < \infty$.

- Each row of Π contains a finite number of nonzero entries or, in other words, if the marking is μ at time θ, the set of possible markings at time $\theta + 1$ is finite.

The first two requirements can be easily verified. The third requirement is certainly satisfied if $S_{s,s'}$ does not contain arbitrarily long sequences. This requirement does not allow to analyze exactly, for example, the DDP-SPN in Fig. 3. However, this behavior can be approximated arbitrarily well using a truncated geometric distribution for the size of the batch arrivals. Incidentally, we observe that the continuous version of this SPN, where t and y are exponentially distributed, shows that Proposition 1 in [9] does not hold for unbounded systems: there is no SPN with only exponentially distributed firing times equivalent to this GSPN (equivalently, there is no SPN with only geometrically distributed firing times equivalent to the DDP-SPN in Fig. 3).

6.1 Truncating the state space

The modified power method algorithm can, in principle, perform the transient analysis of any DDP-SPN that reaches only a finite number of markings (hence states) in a finite amount of time. In practice, though, the number of markings reachable in a finite time might still be too large, hence we need to find ways to reduce the memory requirements.

A first observation allows us to reduce the number of states that must be stored without introducing any approximation. If all the firing times have geometric distributions with parameters less than one, there is a nonzero probability of remaining in a state s for an arbitrary number of steps, once s is entered. Indeed, the assumption of our modified power method algorithm, and of [16, 18], is that the set of explored states never decreases: $\mathcal{S}^{[0]} \subseteq \mathcal{S}^{[1]} \subseteq \mathcal{S}^{[2]} \subseteq \cdots$.

However, some firing times might have distributions with finite support, so it is possible that $\pi_s^{[n]} > 0$ while $\pi_s^{[n+1]} = 0$ and, in this case, state s can be discarded before computing $\mathcal{S}^{[n+2]}$. Then, we can redefine $\mathcal{S}^{[n]}$ to be the set of *time-reachable* states at step n, that is, the states having a nonzero probability at step n: $\mathcal{S}^{[n]} = \{s : \pi_s^{[n]} > 0\}$.

$\mathcal{S}^{[0]}$ is completely determined by $\pi^{[0]}$, which is given, and $\pi^{[n]}$, hence $\mathcal{S}^{[n]}$, is obtained from $\pi^{[n-1]}$ by computing $\Pi_{s,\bullet}$ for each $s \in \mathcal{S}^{[n-1]}$, and then restricting the usual matrix-vector multiplication $\pi^{[n]} = \pi^{[n-1]}\Pi$ to the entries corresponding to $\mathcal{S}^{[n-1]}$, since the other entries are zero anyway. Extreme cases are illustrated in Fig. 6, where, in (a), $\mathcal{S}^{[n]} = \{(1j, 1) : 0 \leq j \leq n\}$, while, in (b), $\mathcal{S}^{[n]} = \{(1n, 1)\}$.

Hence, if a state s is time-reachable at step n, but time-unreachable at step $n + 1$, we can destroy it and its corresponding row in Π at the end of step $n + 1$. At worst, the same state s might become time-reachable again at a later step, and the algorithm will have to compute the corresponding row $\Pi_{s,\bullet}$ for the transition probability matrix again.

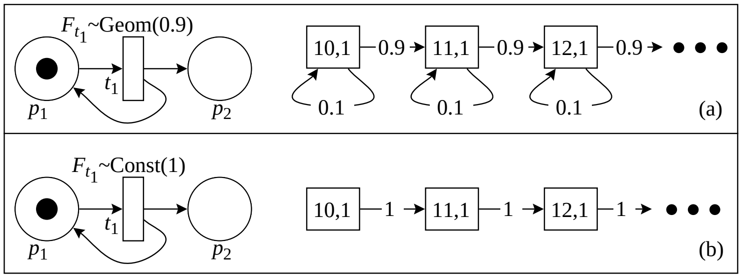

Figure 6: A case where $\mathcal{S}^{[n]} \subseteq \mathcal{S}^{[n+1]}$ and another where $\mathcal{S}^{[n]} \not\subseteq \mathcal{S}^{[n+1]}$.

The observation that the $\mathcal{S}^{[n]}$ are not required to be a sequence of nondecreasing subsets is, we believe, new. Unfortunately, geometric distributions with parameter less than one are often used in practice, resulting in an increasingly larger set of states to be stored at each step of the modified power method.

Further observing that some markings might have negligible probability, however, allows us to avoid keeping $\mathcal{S}^{[n]}$ in its entirety, at the cost of an approximate solution, but with computable bounds. For example, in Fig. 6(a), the probability of marking $(1k, 1)$ at step n, $k \leq n$, is $\binom{n}{k} 0.9^k 0.1^{n-k}$, which is extremely small when the difference between n and k is large. An approximate solution approach based on truncation of the state-space might then be appropriate. At step n, only the states in $\hat{\mathcal{S}}^{[n]} \subseteq \mathcal{S}^{[n]}$ are considered. For each state $s \in \hat{\mathcal{S}}^{[n]}$, its computed probability $\hat{\pi}^n_s$ is an approximation of the exact probability $\pi^{[n]}_s$ at step n:

1. Initially, $\pi^{[0]}$ is known, so set
$$\hat{\pi}^{[0]} \leftarrow \pi^{[0]} \qquad \text{and} \qquad \hat{\mathcal{S}}^{[0]} \leftarrow \{s : \hat{\pi}^{[0]}_s > 0\}.$$
The "total known probability mass" and the "total known sojourn time" at the beginning are
$$\kappa^{[0]} \leftarrow ||\hat{\pi}^{[0]}||_1 = \sum_{s \in \hat{\mathcal{S}}^{[0]}} \hat{\pi}^{[0]}_s = 1 \qquad \text{and} \qquad K^{[0]} \leftarrow 0.$$

2. As the iteration progresses, the size of $\hat{\mathcal{S}}^{[n]}$ might grow too large and states with probability below a threshold c must be truncated, destroying them and their corresponding row in Π, de facto setting their probability to zero
$$\text{for each } s \in \hat{\mathcal{S}}^{[n]} \text{ do} \qquad \text{if } \hat{\pi}^{[n]}_s < c \text{ then} \qquad \hat{\pi}^{[n]}_s \leftarrow 0.$$
Compute the new set of kept states
$$\hat{\mathcal{S}}^{[n]} \leftarrow \{s : \hat{\pi}^{[n]}_s > 0\}.$$

Regardless of whether truncation is performed, the total known probability mass at step n and the total known sojourn time up to step n are

$$\kappa^{[n]} \leftarrow ||\hat{\pi}^{[n]}||_1 = \sum_{s \in \hat{\mathcal{S}}^{[n]}} \hat{\pi}_s^{[n]} \leq 1 \qquad \text{and} \qquad K^{[n]} \leftarrow K^{[n-1]} + \kappa^{[n]} \leq n.$$

Without other information, we can only say that the probability of being in state $s \in \hat{\mathcal{S}}^{[n]}$ at step n is at least $\hat{\pi}_s^{[n]}$, while we do not know how the unaccounted probability mass $\kappa^{[0]} - \kappa^{[n]}$ should be redistributed (we know that it should be redistributed over the states in $\mathcal{S}^{[n]}$, hence some of it could be over states in $\hat{\mathcal{S}}^{[n]} \subseteq \mathcal{S}^{[n]}$, but we have no way to tell). An analogous interpretation holds for $K^{[n]}$.

3. Truncation can be performed as many times as needed, although every application reduces the value of $\kappa^{[n]}$, thus increases our uncertainty about the state of the system.

4. Upon reaching time θ_F, we know that, with probability at least $\kappa^{[\lfloor \theta_F \rfloor]}$, the system is in one of the non-truncated states $\hat{\mathcal{S}}^{[\lfloor \theta_F \rfloor]}$. Conversely, a total of

$$\overline{K}(\theta_F) \leftarrow \theta_F - K^{[\lfloor \theta_F \rfloor]} + \kappa^{[\lfloor \theta_F \rfloor]}(\theta_F - \lfloor \theta_F \rfloor)$$

sojourn time units are unaccounted for. Hence, assuming that the reward rates associated to the states have an upper and lower bound ρ_L and ρ_U, $E[Y(\theta_F)]$ and $E[y(\theta_F)]$ can be bounded as well. If $E[\hat{Y}(\theta_F)]$ and $E[\hat{y}(\theta_F)]$ are the approximations obtained using our truncation approach,

$$E[\hat{y}(\theta_F)] + \rho_L(1 - \kappa^{[\lfloor \theta_F \rfloor]}) \;\leq\; E[y(\theta_F)] \;\leq\; E[\hat{y}(\theta_F)] + \rho_U(1 - \kappa^{[\lfloor \theta_F \rfloor]}),$$

$$E[\hat{Y}(\theta_F)] + \rho_L \overline{K}(\theta_F) \;\leq\; E[Y(\theta_F)] \;\leq\; E[\hat{Y}(\theta_F)] + \rho_U \overline{K}(\theta_F).$$

Highly-reliable systems are particularly good candidates for this state-space truncation, since they have a large number of low-probability states.

6.2 Embedding the DTMC

When performing steady-state analysis, it is possible to perform an embedding of the DTMC, observing it only when particular state-to-state transitions occur. For a simple example, consider the DTMC in Fig. 5, which has a transition from state $(000111, \bullet\bullet\bullet 1)$ to state $(111000, 132\bullet)$ with probability one. If the firing time of transition t_4 were changed to Const(7), instead of Const(1), the DTMC would have to contain six additional states, $(000111, \bullet\bullet\bullet 7)$ through $(000111, \bullet\bullet\bullet 2)$. This is obviously undesirable, and it can be easily avoided by an embedding. The DTMC of the embedded process is exactly that of Fig. 5, we must simply set the expected holding time h_s of each state s to one, except that of $(000111, \bullet\bullet\bullet 1)$, which is set to seven. Then, we can solve the embedded DTMC for steady state and obtain a steady-state probability vector $\tilde{\pi}$ for the embedded process. The steady-state probability vector of the actual process is then obtained by weighting $\tilde{\pi}$ according to the holding times, a well known result applicable to the steady-state solution of any semi-Markov process [8]: $\pi_s = \tilde{\pi}_s h_s \left(\sum_{u \in \mathcal{S}} \tilde{\pi}_u h_u\right)^{-1}$.

For transient analysis, the same idea can be applied, but in a much more restricted way. If, at step n, every state in $\mathcal{S}^{[n]}$ is such that the minimum time before a change of marking is $k > 1$, we can effectively perform an embedding. In the modified power method algorithm, this requires advancing n by k instead of just one step in the outermost loop and adjusting the increment of Y in statement 3 accordingly. It should be noted, however, that this situation is unlikely to occur, since the set $\mathcal{S}^{[n]}$ may contain many states $s = (\mu, \phi)$, and, for each of them, the DTMC describing the firing time of each enabled transition t in μ must satisfy $\min\left\{ l \in \mathbb{N} : \Pr\{\phi_t^{[l]} = 0 \mid \phi_t^{[0]} = \phi_t\} > 0 \right\} \geq k$. This is analogous to the requirement for an efficient application of adaptive uniformization [28] and, as stated in the introduction, it is unlikely to happen in general, especially for large values of θ_F.

7 Conclusion and future work

We defined a class of discrete-time distributions which, when used to specify the firing time of the transitions in a stochastic Petri net, ensures that the underlying stochastic process is a DTMC. We then gave conditions under which the transient analysis of this DTMC can be performed even if the state space is infinite. In practice, though, the memory requirements might still be excessive, hence we explored some state-space reduction techniques.

The implementation of a computer tool based on the DDP-SPN formalism is under way. In particular, algorithms for the efficient computation of the rows of the transition probability matrix, $\Pi_{s,\bullet}$, are being explored.

References

[1] V. S. Adve and M. K. Vernon. The influence of random delays on parallel execution times. In *Proc. 1993 ACM SIGMETRICS Conf. on Measurement and Modeling of Computer Systems*, Santa Clara, CA, May 1993.

[2] T. Agerwala. A complete model for representing the coordination of asynchronous processes. Hopkins Computer Research Report 32, Johns Hopkins University, Baltimore, Maryland, July 1974.

[3] M. Ajmone Marsan, G. Balbo, and G. Conte. A class of Generalized Stochastic Petri Nets for the performance evaluation of multiprocessor systems. *ACM Trans. Comp. Syst.*, 2(2):93–122, May 1984.

[4] M. Ajmone Marsan and V. Signore. Timed Petri nets performance models for fiber optic LAN architectures. *Proc. 2rd Int. Workshop on Petri Nets and Performance Models (PNPM'87)*, pages 66–74, 1987.

[5] R. Y. Al-Jaar and A. A. Desrochers. Modeling and analysis of transfer lines and production networks using Generalized Stochastic Petri Nets. In *Proc. 1988 UPCAEDM Conf.*, pages 12–21, Atlanta, GA, June 1988.

[6] K. Barkaoui, G. Florin, C. Fraize, B. Lemaire, and S. Natkin. Reliability analysis of non repairable systems using stochastic Petri nets. In *Proc. 18th Int. Symp. on Fault-Tolerant Computing*, pages 90–95, Tokyo, Japan, June 1988.

[7] J. Bechta Dugan and G. Ciardo. Stochastic Petri net analysis of a replicated file system. *IEEE Trans. Softw. Eng.*, 15(4):394–401, Apr. 1989.

[8] E. Çinlar. *Introduction to Stochastic Processes.* Prentice-Hall, 1975.

[9] G. Chiola, S. Donatelli, and G. Franceschinis. GSPNs versus SPNs: what is the actual role of immediate transitions? In *Proc. 4th Int. Workshop on Petri Nets and Performance Models (PNPM'91)*, Melbourne, Australia, Dec. 1991. IEEE Computer Society Press.

[10] H. Choi and K. S. Trivedi. Approximate performance models of polling systems using stochastic Petri nets. In *Proc. IEEE INFOCOM 92*, pages 2306–2314, Florence, Italy, May 1992.

[11] G. Ciardo. *Analysis of large stochastic Petri net models.* PhD thesis, Duke University, Durham, NC, 1989.

[12] G. Ciardo, A. Blakemore, P. F. J. Chimento, J. K. Muppala, and K. S. Trivedi. Automated generation and analysis of Markov reward models using Stochastic Reward Nets. In C. Meyer and R. J. Plemmons, editors, *Linear Algebra, Markov Chains, and Queueing Models*, volume 48 of *IMA Volumes in Mathematics and its Applications*, pages 145–191. Springer-Verlag, 1993.

[13] G. Ciardo, R. German, and C. Lindemann. A characterization of the stochastic process underlying a stochastic Petri net. *IEEE Trans. Softw. Eng.*, 20(7):506–515, July 1994.

[14] G. Ciardo and C. Lindemann. Analysis of deterministic and stochastic Petri nets. In *Proc. 5th Int. Workshop on Petri Nets and Performance Models (PNPM'93)*, pages 160–169, Toulouse, France, Oct. 1993. IEEE Computer Society Press.

[15] A. Cumani. ESP - A package for the evaluation of stochastic Petri nets with phase-type distributed transitions times. In *Proc. Int. Workshop on Timed Petri Nets*, Torino, Italy, July 1985.

[16] E. de Souza e Silva and P. Mejía Ochoa. State space exploration in Markov models. In *Proc. 1992 ACM SIGMETRICS Conf. on Measurement and Modeling of Computer Systems*, pages 152–166, Newport, RI, USA, June 1992.

[17] W. K. Grassmann. Means and variances of time averages in Markovian environments. *Eur. J. Oper. Res.*, 31(1):132–139, 1987.

[18] W. K. Grassmann. Finding transient solutions in Markovian event systems through randomization. In W. J. Stewart, editor, *Numerical Solution of Markov Chains*, pages 357–371. Marcel Dekker, Inc., New York, NY, 1991.

[19] M. Holliday and M. Vernon. A Generalized Timed Petri Net model for performance analysis. In *Proc. Int. Workshop on Timed Petri Nets*, Torino, Italy, July 1985.

[20] G. Horton and S. T. Leutenegger. A multi-level solution algorithm for steady state Markov chains. In *Proc. 1994 ACM SIGMETRICS Conf. on Measurement and Modeling of Computer Systems*, pages 191–200, Nashville, TN, May 1994.

[21] A. Jensen. Markoff chains as an aid in the study of Markoff processes. *Skand. Aktuarietidskr.*, 36:87–91, 1953.

[22] J. G. Kemeny and J. L. Snell. *Finite Markov Chains.* D. Van Nostrand-Reinhold, New York, NY, 1960.

[23] C. Lindemann, G. Ciardo, R. German, and G. Hommel. Performability modeling of an automated manufacturing system with deterministic and stochastic Petri nets. In *Proc. 1993 IEEE Int. Conf. on Robotics and Automation*, pages 576–581, Atlanta, GA, May 1993. IEEE Press.

[24] B. Melamed and M. Yadin. Randomization procedures in the computation of cumulative-time distributions over discrete state Markov rocesses. *Operations Research*, 32(4):926–944, July-Aug. 1984.

[25] M. K. Molloy. *On the integration of delay and throughput measures in distributed processing models.* PhD thesis, UCLA, Los Angeles, CA, 1981.

[26] S. Natkin. Reseaux de Petri stochastiques. These de docteur ingeneur, CNAM-Paris, Paris, France, June 1980.

[27] A. L. Reibman and K. S. Trivedi. Numerical transient analysis of Markov models. *Computers and Operations Research*, 15(1):19–36, 1988.

[28] A. P. A. van Moorsel and W. H. Sanders. Adaptive uniformization. *Stochastic Models*, 10(3), 1994.

[29] R. Zijal and R. German. A new approach to discrete time stochastic Petri nets. In *Proc. 11th Int. Conf. on Analysis and Optimization of Systems, Discrete Event Systems*, pages 198–204, Sophia-Antipolis, France, June 1994.

[30] W. M. Zuberek. D-timed Petri nets and the modeling of timeouts and protocols. *Trans. of the Society for Computer Simulation*, 4(4):331–357, 1987.