Deriving the performance indices in product-form stochastic Petri nets: open problems and simulation

Simonetta Balsamo, Andrea Marin, Ivan Stojic
Università Ca' Foscari Venezia
DAIS
email: {balsamo,marin,stojic}@dais.unive.it

KEYWORDS
Stochastic Petri nets, Simulation, Product-form

ABSTRACT

Stochastic Petri nets are an important formalism used for the performance evaluation of computer and communication systems as well as other fields like bioinformatics and logistics. Despite its high flexibility and modelling power, one of the problems of quantitative analyses based on stochastic Petri nets is the state space explosion, i.e., the high cardinality reached by the state space of even a structurally small SPN. As a consequence a direct analysis of the Markovian processes underlying the models is not feasible. Product-form Petri nets are a class of stochastic Petri nets whose invariant measure can be expressed as a product of functions, each of which depends only on a marking of a single place. Nevertheless, for the effective computation of the performance indices the computation of the stationary distribution is required. In this paper we propose a classification of product-form stochastic Petri nets based on the availability of algorithms for the computation of their stationary performance indices. Moreover, in case simulation is required, we introduce two stopping criteria that exploit the product-form property of the nets.

INTRODUCTION

Stochastic Petri nets (SPNs) - Molloy (1982), Marsan et al. (1995) - are an important formalism for assessing the performances of computer and telecommunication systems. More recently, they have been used also in other domains, such as bioinformatics with the aim of modelling biochemical reactions in organic systems (see Baldan et al. (2010) for a recent survey). SPNs, as defined in Molloy (1982), are a Markovian modelling formalism, in the sense that the stochastic process underlying the marking process (the process describing evolution of the model's state with time) is a Continuous Time Markov Chain (CTMC). Markovian formalisms are highly appreciated because they can be analysed by numerous techniques and algorithms which have been developed for studying Markov processes in the latest decades. However, SPNs share with many other high level formalisms the problem of the state space explosion. Namely, an SPN which is structurally "small" may have a huge, possibly infinite, state space. As a consequence, although in principle the standard algorithm for transient or steady-state analysis of CTMCs could be applied, in practice time and space complexity become prohibitive and problems concerning numerical stability often arise.

In order to overcome these problems, product-form models have been introduced first in queueing networks (see Jackson (1963), Baskett et al. (1975)) and then in other formalisms including SPNs as proposed in Mairesse and Nguyen (2009), Balsamo et al. (2012), Marin et al. (2012). Product-form analyses rely on the idea that the model can be decomposed into a set of interacting components. When certain conditions are satisfied, each positive recurrent state of the underlying CTMC has a stationary equilibrium probability that can be expressed as a product of equilibrium probabilities of corresponding states of model components, obtained by considering the components in isolation. Thus, product-forms allow analyses to be performed by studying isolated components, and the solution of the system of global balance equations of the CTMC underlying the model is not required anymore. Nevertheless, for SPNs there are still some problems to address before deriving the stationary performance indices. The first problem concerns identification of the reachability set of the SPN, needed to identify the set of aforementioned positive recurrent states of the underlying CTMC. In some cases (e.g., marked graphs or state machines) the problem of deciding whether a state is reachable is computationally efficient, but for general SPNs this problem is known to belong to the class of EXPSPACE problems. Moreover, structural conditions imposed on SPN models by the product-form analysis do not reduce this complexity - Haddad et al. (2013).

In this paper we review open problems concerning the effective computation of some stationary performance indices for product-form SPNs. We show that for some SPNs, at the state of the art, one has to resort to stochastic simulation in order to obtain the desired indices, and we discuss how it is possible to exploit the product-form property to define an efficient criterion to stop the simulation.

STOCHASTIC PETRI NETS

In this section we give the definition of Stochastic Petri Nets (SPNs) and introduce notation used in the paper. An SPN Molloy (1982) is a 6-tuple

$$SPN = (\mathcal{P}, \mathcal{T}, I(\cdot), O(\cdot), W(\cdot), \mathbf{m}_0)$$

where $\mathcal{P} = \{P_1, \ldots, P_{N_P}\}$ is the set of $N_P > 0$ places, $\mathcal{T} = \{T_1, \ldots, T_{N_T}\}$ is the set of $N_T > 0$ transitions, $I : \mathcal{T} \to \mathbb{N}^{N_P}$ is a function associating an input vector with each transition $T_i \in \mathcal{T}$ and $O : \mathcal{T} \to \mathbb{N}^{N_P}$ is a function associating an output vector with each transition, $\mathbf{m}_0 \in \mathbb{N}^{N_P}$ is called initial marking of the net. Function $W : \mathcal{T} \to \mathbb{R}^+$ assigns a positive real number to each transition $T_i \in \mathcal{T}$. A transition T_i is enabled by a marking $\mathbf{m} \in \mathbb{N}^{N_P}$ if $\mathbf{m} - I(T_i) \geq \mathbf{0}$, i.e., has only non-negative components. We define enabling degree of a transition T_i in marking $\mathbf{m}$ by $e_i(\mathbf{m}) = \max\{k \in \mathbb{N} : \mathbf{m} - kI(T_i) \geq \mathbf{0}\}$. In general, a marking $\mathbf{m}$ enables zero, one or more transitions. Let $\mathcal{E}(\mathbf{m})$ be the set of transitions enabled by marking $\mathbf{m}$. When a transition $T_i \in \mathcal{E}(\mathbf{m})$ fires, the marking changes from $\mathbf{m}$ to $\mathbf{m} - I(T_i) + O(T_i)$, i.e., the tokens specified by the input vector are consumed and those specified by the output vector are produced. In Markovian Petri nets (or simply SPNs), we associate an exponentially distributed random delay with each transition enabled by a marking $\mathbf{m}$. Thus, the non-determinism on standard Petri nets is solved with the *race policy* among exponential distributions. We consider two firing semantics:

- *Single server semantics:* in this case a firing delay is set when the transition is first enabled and a new delay is sampled in case the same transition is enabled after a firing. In other words, the firing rate of enabled transition T_i is state independent and its value is $W(T_i)$;

- *Infinite server semantics:* in this case every enabling set of tokens is processed in parallel as soon as they arrive at the input places. Each of these concurrent delays associated with transition T_i are i.i.d. exponentially distributed random variables with rate $W(T_i)$. According to the race policy, this corresponds to a single server semantics in which the firing rate depends on the marking in the transition's input places. More formally, the firing rate of transition T_i in marking $\mathbf{m}$ is $e_i(\mathbf{m})W(T_i)$.

Given the initial marking $\mathbf{m}_0$, set $RS(\mathbf{m}_0)$ is the set of all the possible markings reachable after an arbitrary number of transition firings from $\mathbf{m}_0$. Reachability graph of an SPN has the elements of $RS(\mathbf{m}_0)$ as nodes and the arcs connect markings which are reachable via the firing of a transition (directly reachable markings). The marking process is the stochastic process $X(t)$ associated with the evolution of the net's marking for $t \in \mathbb{R}_{\geq 0}$. It can be proved that for SPNs, $X(t)$ is a Continuous Time Markov Chain (CTMC) whose transition graph structure is identical to that of the SPN's reachability graph. The transition rates are set according to the definition of function $W(\cdot)$ and the firing semantics which is adopted. The derivation of the reachability graph of an SPN is known to belong to the class of EXPSPACE problems. Incidence matrix $\mathbf{A}$ of an SPN is a matrix which has a row for each place and a column for each transition. The column associated with transition T_i is $I(T_i) - O(T_i)$, and represents the marking change due to firing of T_i. A T-invariant for an SPN is a vector $\mathbf{X} > \mathbf{0}$ whose dimension is equal to the number of net's transitions, such that:

$$\mathbf{A} \cdot \mathbf{X} = \mathbf{0}\,,$$

where $\mathbf{X} = (x_1, \ldots, x_{N_T})^T$ and $x_i \in \mathbb{N}$. A P-invariant of the net is a vector $\mathbf{Y} > \mathbf{0}$ whose dimension is equal to the number of places of the SPN, and which satisfies the following equation:

$$\mathbf{A}^T \cdot \mathbf{Y} = \mathbf{0}\,,$$

where $\mathbf{Y} = (y_1, \ldots, y_{N_P})^T$ and $y_i \in \mathbb{N}$. Support of a P-invariant $\mathbf{Y}$ is the set of places for which corresponding components of $\mathbf{Y}$ are nonzero, and support is minimal if there are no P-invariants with smaller (in terms of subset) support. For each minimal support there is a unique P-invariant called minimal support P-invariant; set of minimal support P-invariants forms a basis for all P-invariants of an SPN.

Some performance indices in SPNs

In this part we review some performance measures for SPNs in equilibrium. Henceforth we assume that the CTMCs underling the SPNs we consider are ergodic. Let π be the function that assigns its equilibrium probability to each positive recurrent state of the CTMC underlying the SPN. Then, the expected number of tokens in place P_i in steady-state is given by the following expression:

$$\overline{N}_{P_i} = \sum_{\mathbf{m}} \pi(\mathbf{m}) m_i\,, 1 \leq i \leq N_P\,. \tag{1}$$

The throughput of a single server transition T_i in steady-state is given by:

$$\overline{X}_{T_i} = \sum_{\mathbf{m}} \pi(\mathbf{m}) \delta_{e_i(\mathbf{m}) \geq 0} W(T_i)\,, \tag{2}$$

with $1 \leq i \leq N_T$ and δ is the indicator function. For infinite server semantics the throughput is given by:

$$\overline{X}_{T_i} = \sum_{\mathbf{m}} \pi(\mathbf{m}) e_i(\mathbf{m}) W(T_i)\,. \tag{3}$$

Product-form stochastic Petri nets

A subclass of SPNs are known to be in product-form, i.e., the expression of the equilibrium distribution of the net's marking process is such that:

$$\pi(\mathbf{m}) = \frac{1}{G}\prod_{i=1}^{N_P} g_i(m_i)\,, \tag{4}$$

where $\mathbf{m}$ is a positive recurrent state of the CTMC, π is the equilibrium probability function, G is the normalising constant such that the probabilities sum to 1, and g_i are some positive real functions.
For the sake of simplicity we briefly review the results on product-form SPNs only for nets whose transitions have the single server semantics. According to Coleman et al. (1996) a large class of SPNs in product-form satisfies the following conditions:

1. Let $\mathcal{I}$, $\mathcal{O}$ be the sets of the input and output vectors of the net transitions, respectively. Then, $\mathcal{I} = \mathcal{O}$.
2. No two transitions have the same input vector, i.e. $i \neq k \Rightarrow I(T_i) \neq I(T_k)$. Nets which don't satisfy this condition can be modified by considering each set of transitions that share an input vector as a compound transition, in the following manner. If more than one transition in the net has the same input vector $I(T_i)$, we replace the set $\{T_k : I(T_k) = I(T_i)\}$ of these transitions with a compound transition T and we set $W(T) = \sum_{k:I(T_k)=I(T_i)} W(T_k)$. Firing of the compound transition T in the modified net represents firing of one of the associated original transitions in the original net. When the compound transition T fires, one of the output vectors of the associated original transitions is selected probabilistically. The selection probabilities are derived based on the properties of the exponential distribution so as to preserve the underlying CTMC of the original net. Thus, we set probability $p(I(T_i), O(T_j))$ of generating the tokens specified by $O(T_j)$ when the compound transition T fires to be equal to the probability of firing in the original net the transition T_j given that one of the transitions with the input vector $I(T_i)$ fires:

   $$p(I(T_i), O(T_j)) = \frac{W(T_j)}{W(T)} = \frac{W(T_j)}{\sum_{k:I(T_k)=I(T_i)} W(T_k)}\,.$$

   The discrete-time Markov chain with state space $\mathcal{I}$ and the above transition probabilities is called the routing process.
3. There exists an invariant measure $f : \mathcal{I} \to \mathbb{R}^+$ of the routing process such that:

   $$\chi(i)f(i) = \sum_{j\in\mathcal{I}} \chi(j)f(j)p(j,i)\,,$$

   where $\chi(k) = \sum_{T:I(T)=k} W(T)$.

Theorem 1 *Let $\mathbf{C}(f)$ be the vector whose components correspond to the transitions, and let the component associated with T_i be equal to $\log(f(I(T_i))/f(O(T_i)))$. Since all invariant measures of the routing process differ by a positive multiplicative constant we can simply write $\mathbf{C}(f)$ as $\mathbf{C}$. Then, if the equation*

$$-\mathbf{A}\begin{bmatrix}\log(y_1)\\ \vdots \\ \log(y_{N_P})\end{bmatrix} = \mathbf{C}$$

has a unique solution then, under ergodicity assumption, for each positive recurrent state $\mathbf{m}$ it holds that

$$\pi(\mathbf{m}) = \frac{1}{G}\prod_{i=1}^{N_P} y_i^{m_i}\,,$$

where G is the normalising constant.

In the literature, several other classes of SPNs in product-form have been proposed such as that based on Boucherie's full-blocking Lazar and Robertazzi (1991), Boucherie (1994), signals in the style of G-networks Marin et al. (2012) and others Balbo et al. (2003), Balsamo and Marin (June, 2007;O).

DERIVING THE PERFORMANCE INDICES FOR PRODUCT-FORM SPNS

Although Theorem 1 gives the expression for the unnormalized equilibrium distribution for a class of SPNs, derivation of the stationary performance indices requires knowledge of the normalized equilibrium probability distribution. Therefore, the efficiency of the product-form approach strongly relies on the capability of computing the normalising constant G efficiently. In this section we distinguish three classes of product-form SPNs based on applicable methods for computing the normalising constant (or possibly directly the performance indices).

Cartesian product-form SPNs

This class of product-form SPNs satisfies the property that the set $RS(\mathbf{m}_0)$ of markings reachable from an initial marking $\mathbf{m}_0$ is Cartesian product over places of reachable markings of places. More formally, for place indices $i \in \{1,\ldots,N_P\}$, let $S_i(\mathbf{m}_0) = \{k : \exists \mathbf{m} \in RS(\mathbf{m}_0) \text{ s.t. } m_i = k\}$ be the reachable markings of places. Then, an SPN is in Cartesian product-form if it satisfies the conditions of Theorem 1 and if

$$RS(\mathbf{m}_0) = S_1(\mathbf{m}_0) \times S_2(\mathbf{m}_0) \times \cdots \times S_{N_P}(\mathbf{m}_0)\,.$$

For this class of SPNs, we can define for each place P_i a constant $G_i = \sum_{k\in S_i(\mathbf{m}_0)} y_i^k$. Then, it is easy to see that by Equation (4) we have:

$$G = \prod_{i=1}^{N_P} G_i\,.$$

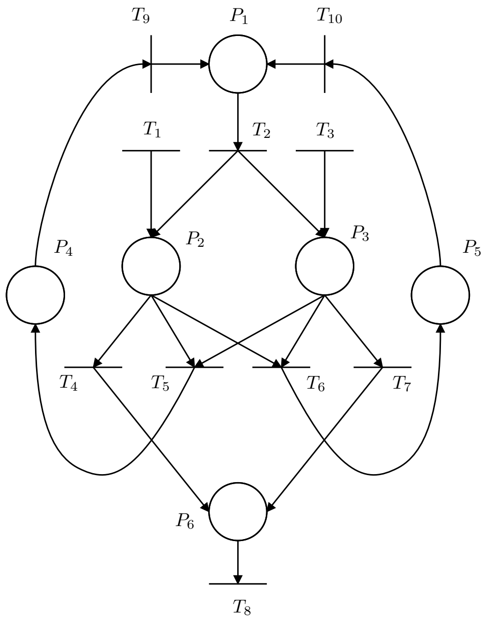

Figure 1: Example of Cartesian product-form SPN.

As a consequence, the performance indices can be readily derived. For those who are familiar with queueing network theory, this is the case for Jackson's networks and G-networks - Jackson (1963), Gelenbe (1989). For this class of models, each queueing stations can have a state in $\mathbb{N}$ and the joint process state space is $\mathbb{N}^Q$, with Q being the number of the network's stations. Figure 1 shows an example of this class of SPNs: customers arrive from the outside via T_1 and T_3 at places P_2 and P_3. The service of customers can cause a collision (T_5 or T_6) causing the customers to be kept in an idle phase (P_4, P_5) and then put newly in service (firing of transition T_2). Each place has a marking which belongs to $\mathbb{N}$ and the joint state space is $\mathbb{N}^6$.

P-invariant reachable product-form SPNs

This class of SPNs is characterised by the fact that deciding if a marking belongs to the reachability set can be performed in polynomial time. In fact, we have that a marking is reachable if and only if given a matrix $\mathbf{M}$ of minimal support P-invariants, a necessary and sufficient condition for the reachability of any marking $\mathbf{m}$ is:

$$\mathbf{Mm} = \mathbf{Mm}_0 \,. \qquad (5)$$

This class of product-form SPNs has been introduced in Coleman et al. (1996) and further studied in Sereno and Balbo (1997). For these models, in Coleman et al. (1996) the authors propose a convolution algorithm while in Sereno and Balbo (1997) a mean value analysis algorithm is proposed. The main problem for the computation of the performance indices of this class of SPNs is how to decide if the property stated by Equation (5) holds. Indeed, in general one has to derive the reachability set of the net, derive the minimal support P-invariants and verify Equation (5) for the matrix $\mathbf{M}$ of minimal support P-invariants; alternatively one can resort to some model-dependent proof technique. An example of P-invariant reachable SPN taken from Coleman et al. (1996) is shown in Figure 2.

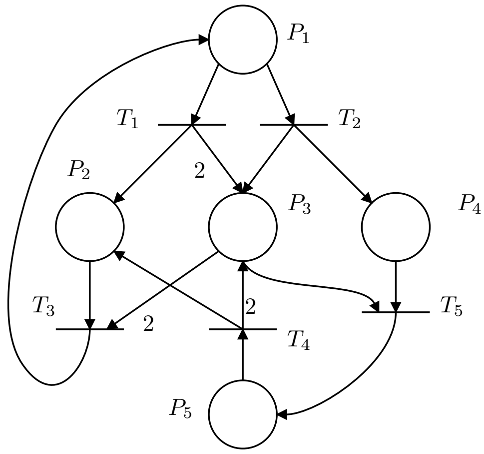

Figure 2: Example of P-invariant reachable product-form SPN.

General product-form SPNs

If the product-form SPN does not belong to any of the above mentioned classes, the only possibility for carrying out an exact analysis is to derive the whole reachability set and evaluate Equation (4) for each positive recurrent state. If the state space is finite then the normalisation of the probabilities can be performed. Unfortunately, this method tends to be prone to numerical instability problems and it is time/space expensive because it requires the construction of the whole state space whose complexity is EXPSPACE.

SIMULATION OF PRODUCT-FORM SPNs

Although resorting to simulation for product-form SPNs seems to be a contradiction with the analytical properties of the product-form models, in some cases it is the only possible choice. Summing up, the conditions under which one should consider to use the simulation to study a product-form SPN are:

1. The SPN is neither Cartesian nor P-invariant reachable or proving that one of these properties holds is computationally expensive;

2. The state space of the SPN is so large to make the brute force approach infeasible. By brute force approach we mean the application of Equation (4) for each positive recurrent state and then normalising

the probabilities.

In this section we discuss two possibilities for defining a stopping criterion for the simulation of product-form SPNs and we test their performance with a simulator implemented in Java.

Stopping criteria for the simulation of product-form SPNs

In the computation of the stationary performance indices, it is important to choose the length of two periods: the first period is delimited by the epoch at which the transient period expires and hence we can consider the model in its stationary behaviour. The second is the minimum epoch at which the simulation can be stopped in order to have accurate estimates of the desired performance indices. We have addressed the first problem by proposing a perfect sampling approach for SPNs in Balsamo et al. (2015) but several other methods developed in the literature can be applied (e.g., the Welch's procedure presented in Welch (1983)).
In this paper we focus on the definition of a stopping criterion in the simulation of product-form SPNs. Standard stopping criteria usually rely on carrying out the stationary simulation until a certain level of accuracy of the estimated performance measure is achieved. For instance, a typical approach is defining a confidence interval and a maximum tolerance on an average measure such as the expected number of tokens in a certain SPN's place. Intuitively, longer simulations should decrease the variance of the estimation and hence reduce the width of the confidence intervals until the desired accuracy is reached. The drawback of this approach is that it may be difficult to apply for the estimation of the probability of rare or unlikely events.

Matrix based stopping criterion

In the simulation of product-form SPNs we exploit the fact that although we do not know the probability of an arbitrary reachable state (because of the unknown normalising constant), from Equation (4) we can derive exactly the ratio between probabilities of two arbitrary states. Assume that we have a finite set of reachable and ergodic markings

$$\mathcal{U} = \{\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_U\},$$

and let $\tilde{\pi}_i$ be the estimation of the stationary probability for marking $\mathbf{m}_i$ in a stochastic simulation of the SPN. Then, we define the matrix $\tilde{\mathbf{U}}$ as follows:

$$\tilde{\mathbf{U}} = \begin{bmatrix} \frac{\tilde{\pi}_1}{\tilde{\pi}_1} & \frac{\tilde{\pi}_2}{\tilde{\pi}_1} & \dots & \frac{\tilde{\pi}_U}{\tilde{\pi}_1} \\ \frac{\tilde{\pi}_1}{\tilde{\pi}_2} & \frac{\tilde{\pi}_2}{\tilde{\pi}_2} & \dots & \frac{\tilde{\pi}_U}{\tilde{\pi}_2} \\ & \vdots & \vdots & \vdots \\ \frac{\tilde{\pi}_1}{\tilde{\pi}_U} & \frac{\tilde{\pi}_2}{\tilde{\pi}_U} & \dots & \frac{\tilde{\pi}_U}{\tilde{\pi}_U} \end{bmatrix}. \tag{6}$$

We can derive the analytical values for the ratios of $\tilde{\mathbf{U}}$ thanks to the product-form property of the SPN. Let matrix $\mathbf{U}$ be defined as:

$$\mathbf{U} = \begin{bmatrix} \frac{\pi_1}{\pi_1} & \frac{\pi_2}{\pi_1} & \dots & \frac{\pi_U}{\pi_1} \\ \frac{\pi_1}{\pi_2} & \frac{\pi_2}{\pi_2} & \dots & \frac{\pi_U}{\pi_2} \\ & \vdots & \vdots & \vdots \\ \frac{\pi_1}{\pi_U} & \frac{\pi_2}{\pi_U} & \dots & \frac{\pi_U}{\pi_U} \end{bmatrix}, \tag{7}$$

where π_i denotes an invariant measure of marking $\mathbf{m}_i \in \mathcal{U}$ obtained by Equation (4). Clearly $\tilde{\mathbf{U}}$ changes along the simulation and if the accuracy of the simulation increases we will have that at a certain point $\tilde{\mathbf{U}} \simeq \mathbf{U}$. More formally, we stop the stationary simulation when:

$$|\tilde{\mathbf{U}} - \mathbf{U}| < \varepsilon, \tag{8}$$

where ε is a small positive real number and $|\cdot|$ is the Frobenius' norm.

Vector based stopping criterion

Let $\mathcal{U}$ be a finite set of positive recurrent markings. Then, for each $\mathbf{m}_i \in \mathcal{U}$ we have:

$$G = \frac{\pi_i}{\prod_{j=1}^{N_P} g_j(m_{i,j})},$$

where π_i is the steady-state probability of $\mathbf{m}_i$, g_j are the functions from the product-form expression (4) and $m_{i,j}$ is the component associated with place P_j in marking $\mathbf{m}_i$. Let $\tilde{\pi}_i$ be the estimation of the stationary probability for marking $\mathbf{m}_i$ in a simulation run and let

$$G_i = \frac{\pi_i}{\prod_{j=1}^{N_P} g_j(m_{i,j})}.$$

The we can define the vector $\tilde{\mathbf{V}}$ as:

$$\tilde{\mathbf{V}} = [G_1, G_2, \dots, G_U].$$

Let $\overline{G}$ be defined as follows:

$$\overline{G} = \frac{\sum_{i=1}^{U} G_i}{U},$$

and let $\mathbf{V}$ be a U-dimensional vector defined as:

$$\mathbf{V} = [\overline{G}, \dots, \overline{G}],$$

Then the stopping criterion is:

$$|\tilde{\mathbf{V}} - \mathbf{V}| < \varepsilon, \tag{9}$$

where ε is a small positive real number and $|\cdot|$ is the Frobenius' norm.

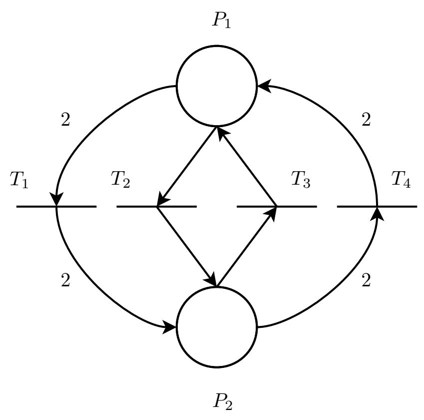

Figure 3: Example of product-form SPN with finite state space.

Determining set $\mathcal{U}$

Set $\mathcal{U}$ can be determined in several ways. Clearly, if we are interested in evaluating the stationary probabilities of some specific markings (e.g., because they are associated with some interesting event), we include those markings in $\mathcal{U}$. However, in general we are interested in including in $\mathcal{U}$ markings with high probability mass. In order to achieve this we perform a random walk in the CTMC underlying the SPN and we keep track of the U reachable markings with highest probabilities. Notice that since the SPN is in product-form, once we visit a marking we can evaluate its stationary probability modulo a multiplicative positive constant since we know its unnormalised stationary probability. Hence, we can compare the stationary probabilities of the reached markings to choose those with highest stationary probabilities.

EXPERIMENTS

In this section we present the outcomes of the simulations for some product-form stochastic Petri nets. The purpose is to validate the stopping criteria introduced in the previous section. In order to achieve this goal we apply the vector and matrix based stopping criteria to different nets which belongs to the class of Cartesian or P-invariant reachable product-form SPNs or whose state space is tractable with the brute force approach. Then, we verify if the analytical values of the performance indices or state probabilities fall in a confidence interval of 95% or 99%.

Finite state space SPN with matrix stopping criterion

Let us consider the SPN depicted in Figure 3. The transition rates are set to 5, 4, 25, 2 for T_1, T_2, T_3 and T_4, respectively. The initial marking is $[6, 4]$. The net is simple and its reachability set consists of only 11 states but it will be useful to validate our approach. We set $|\mathcal{U}| = 4$ and the random walk returns the following markings:

$$\mathcal{U} = \{[1, 9], [3, 7], [0, 10], [2, 8]\}\,.$$

We set $\varepsilon = 1.5{\cdot}10^{-3}$. We performed 50 independent runs where the transient phase has been removed according to the Welch's method. In the average the simulation had to process $235 \cdot 10^3$ transition firings to reach the desired accuracy. For the markings in $\mathcal{U}$ we obtain the following values, where Δ_i represents the width of the confidence interval for the 95% confidence level:

Marking	$\tilde{\pi}_i$	π_i	Δ_i
$[1, 9]$	0.240037	0.240010	$3.56190 \cdot 10^{-4}$
$[3, 7]$	0.038401	0.038401	$4.70516 \cdot 10^{-4}$
$[0, 10]$	0.600029	0.600025	$2.37797 \cdot 10^{-4}$
$[2, 8]$	0.0960075	0.096004	$3.95243 \cdot 10^{-4}$

We observe that all the analytical values of the marking probabilities fall in the confidence interval even though it is very small.

Infinite state space SPN with matrix stopping criterion

We consider the Cartesian product-form SPN depicted in Figure 4 where the rates are 1.0, 5.0, 9.0, 3.0, 7.0, 4.0, 1.7, 3.8 for the transitions $T_1, \ldots, T_8$. The initial marking is $[1, 1, 0, 1, 0]$ and $|\mathcal{U}| = 4$. We perform 50 independent simulation runs. The algorithm constructing set $\mathcal{U}$ returns the following markings:

$$\mathcal{U} = \{[0, 0, 0, 0, 0], [0, 0, 0, 0, 1], [0, 0, 0, 1, 0], [1, 0, 0, 0, 0]\}\,.$$

By setting $\varepsilon = 3 \cdot 10^{-3}$ and a confidence level of 95% we obtain the following estimates for the stationary state probabilities:

Marking	$\tilde{\pi}_i$	π_i	Δ_i
$[0, 0, 0, 0, 0]$	0.011779	0.011772	$4.3 \cdot 10^{-5}$
$[0, 0, 0, 0, 1]$	0.006931	0.006924	$2.3 \cdot 10^{-5}$
$[0, 0, 0, 1, 0]$	0.007235	0.007228	$2.7 \cdot 10^{-5}$
$[1, 0, 0, 0, 0]$	0.007070	0.007063	$2.5 \cdot 10^{-5}$

Finite state space SPN with vector stopping criterion

Let us consider again the SPN depicted in Figure 3. In this case the rates are 10, 1, 25, 2, for $T_1, \ldots, T_4$, respectively and the initial marking is $[5, 3]$. We set $|\mathcal{U}| = 4$ and the random walk returns the following markings:

$$\mathcal{U} = \{[3, 5], [1, 7], [0, 8], [2, 6]\}$$

We performed 50 independent runs where $\varepsilon = 1 \cdot 10^{-6}$. The expected number of processed events to reach the

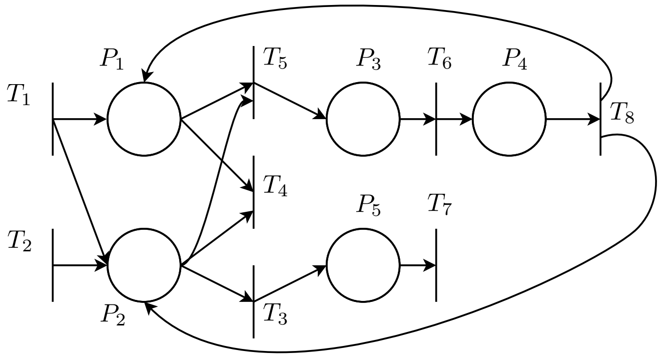

Figure 4: Example of product-form SPN with infinite state space.

desired accuracy are $485 \cdot 10^3$. We obtain the following estimates for the stationary probabilities for a confidence level of 99%:

Marking	$\tilde{\pi}_i$	π_i	Δ_i
$[3,5]$	0.006399	0.00640	$4 \cdot 10^{-6}$
$[1,7]$	0.16005	0.16	$6.1 \cdot 10^{-5}$
$[0,8]$	0.79997	0.8	$5.5 \cdot 10^{-5}$
$[2,6]$	0.031996	0.032	$1.3 \cdot 10^{-5}$

Also in this case all the analytical values for the stationary state probabilities fall in the confidence interval.

Infinite state space SPN with vector stopping criterion

We consider again the SPN of Figure 4 with transition rates 1.0, 5.0, 9.0, 3.0, 7.0, 4.0, 1.7, 3.8 for $T_1, \ldots, T_8$, respectively. We set $\varepsilon = 2 \cdot 10^{-8}$ and $|\mathcal{U}| = 11$. The initial marking is $[1,1,0,1]$. The markings identified by the random walk algorithm are:

$$\mathcal{U} = \{[0,0,1,1,0],[0,0,0,0,0],[0,0,0,1,0],[1,0,0,1,0], [2,0,0,0,0],[0,0,0,2,0],[0,0,1,0,0],[0,0,0,1,1], [0,1,0,0,0],[1,0,0,0,0],[0,0,0,0,1]\}$$

We obtain the following estimates for the stationary probabilities for a confidence level of 95%:

Marking	$\tilde{\pi}_i$	π_i	Δ_i
$[0,0,1,1,0]$	0.004216	0.004216	$1.1E-5$
$[0,0,0,0,0]$	0.011756	0.011772	$3-9E-5$
$[0,0,0,1,0]$	0.007238	0.007228	$2.4E-5$
$[1,0,0,1,0]$	0.004339	0.004337	$1.2E-5$
$[2,0,0,0,0]$	0.004237	0.004238	$1.2E-5$
$[0,0,0,2,0]$	0.004439	0.004438	$1.3E-5$
$[0,0,1,0,0]$	0.006871	0.006867	$2.3E-5$
$[0,0,0,1,1]$	0.004254	0.004252	$1.2E-5$
$[0,1,0,0,0]$	0.006534	0.006540	$2.4E-5$
$[1,0,0,0,0]$	0.007067	0.007063	$2.4E-5$
$[0,0,0,0,1]$	0.006914	0.006924	$2.1E-5$

Also in this case the analytical values for the stationary state probabilities fall in the confidence intervals.

Expected number of tokens in a place for infinite state space SPN

In this section we study again the SPN of Figure 4 but we are interested in estimating the expected number of tokens in P_5 rather then the stationary probability for a marking. We compare the accuracy of the estimation with that provided by Timenet Zimmermann et al. (2000). The Transition rates are 1.5, 3.0, 11.0, 19.0, 9.0, 2.0, 1.6, 3.0 for the transitions T_1, …, T_8, respectively. The initial marking is $[1,1,0,1,0]$. We choose $|\mathcal{U}| = 4$. The analytical value for the expected number of tokens in P_5 is 15. We execute different tests in order to have an approximate relative width of the confidence interval of 5%, 4%, 3%, 2% and 1%. We used the matrix based stopping criterion where the values for ε have been set in order to obtain the same relative width of the confidence interval. As for the product-form simulation we obtain the following table:

Rel. width	Estimate	Rel. Err.
0.05297	15.127683	0.85%
0.04636	14.893279	0.71%
0.03384	14.930198	0.46%
0.02559	14.997131	0.02%
0.01620	15.002821	0.018%

As for the simulation estimates obtain with Timenet we have the following outcomes:

Rel. width	Estimate	Rel. Err.
0.05032	15.188676	1.25%
0.04213	15.156485	1.04%
0.03383	15.097450	0.65%
0.02732	15.006833	0.04%
0.01187	15.002978	0.019%

We observe that the relative error obtained with the stopping criterion based on the product-form property of the SPN is always lower than the relative error obtained with stopping criteria applicable for general SPNs.

CONCLUSIONS

In this paper we have reviewed the problems concerning the effective computation of the performance indices or the stationary state probabilities in product-form stochastic Petri nets. Indeed, although this class of models admits a separable solution for the stationary distribution that potentially allows for an analytical tractability of the performance measures, the problem of the efficient computation of the normalising constant is still open. We identified two classes of product-form SPNs for which determining the normalising constant and hence the stationary performance indices is computationally feasible. For the product-form SPNs that do not belong to this class, if the state space is too large for brute-force normalisation of the probabilities, one has to resort to simulation. To the best of our knowledge, at the state of the art, there does not exist any algorithm that exploits the product-form property of the SPN in its simulation. In this paper we have proposed to use this property in the definition of two criteria for stopping the simulation. We have validated the two criteria on some SPNs and compared the accuracy of the estimates with those obtained by using halting criteria for general SPN. We showed that the proposed criteria improve the accuracy of the estimates. Future works include the application of the proposed approach to estimate the performances of sampling and game-theory based algorithms Albarelli et al. (2011), Torsello et al. (2011).

REFERENCES

Albarelli A.; Rodolà E.; Bergamasco F.; and Torsello A., 2011. *A non-cooperative game for 3D object recognition in cluttered scenes*. In *Proc. of 3DIMPVT 2011*. 252–259.

Balbo G.; Bruell S.C.; and Sereno M., 2003. *On the relations between BCMP Queueing Networks and Product Form Solution Stochastic Petri Nets. Proc of 10th Int Workshop on Petri Nets and Performance Models, 2003*, 103–112.

Baldan P.; Cocco N.; Marin A.; and Simeoni M., 2010. *Petri nets for modelling metabolic pathways: a survey. Natural Computing*, 9, no. 4, 955–989.

Balsamo S.; Harrison P.G.; and Marin A., 2012. *Methodological Construction of Product-Form Stochastic Petri-Nets for Performance Evaluation. J of System and Software*, 85, no. 7, 1520–1539.

Balsamo S. and Marin A., June, 2007. *On representing multiclass M/M/k queues by generalized stochastic Petri nets*. In *Proc. of Int. Conf. ASMTA 2007*. Prague, Czech Republic, 121–128.

Balsamo S. and Marin A., October 23-25, 2007. *Representing LCFSPR BCMP service centers with Coxian service time distribution*. In *Proc. of Valuetools 2007 Conf.* Nantes, France.

Balsamo S.; Marin A.; and Stojic I., 2015. *Perfect sampling in stochastic Petri nets using decision diagrams*. In *Proc. of Int. Conf. MASCOTS 2015*. To appear.

Baskett F.; Chandy K.M.; Muntz R.R.; and Palacios F.G., 1975. *Open, Closed, and Mixed Networks of Queues with Different Classes of Customers. J ACM*, 22, no. 2, 248–260.

Boucherie R.J., 1994. *A Characterisation of independence for competing Markov chains with applications to stochastic Petri nets. IEEE Trans on Software Eng*, 20, no. 7, 536–544.

Coleman J.L.; Henderson W.; and Taylor P.G., 1996. *Product form equilibrium distributions and a convolution algorithm for Stochastic Petri nets. Perf Eval*, 26, no. 3, 159–180.

Gelenbe E., 1989. *Random neural networks with negative and positive signals and product form solution. Neural Computation*, 1, no. 4, 502–510.

Haddad S.; Mairesse J.; and Nguyen H., 2013. *Synthesis and Analysis of Product-form Petri Nets. Fundam Inform*, 122, no. 1-2, 147–172.

Jackson J.R., 1963. *Jobshop-Like Queueing Systems. Management Science*, 10, 131–142.

Lazar A.A. and Robertazzi T.G., 1991. *Markovian Petri Net Protocols with Product Form Solution. Perf Eval*, 12, no. 1, 67–77.

Mairesse J. and Nguyen H.T., 2009. *Deficiency Zero Petri Nets and Product Form*. In *Proc. of the 30th Int. Conf. on App. and Theory of Petri Nets*. Springer-Verlag, Paris, France, PETRI NETS '09, 103–122.

Marin A.; Balsamo S.; and Harrison P., 2012. *Analysis of stochastic Petri nets with signals. Perf Eval*, 85, no. 7, 1520–1539.

Marsan M.A.; Balbo G.; Conte G.; Donatelli S.; and Franceschinis G., 1995. *Modelling with generalized stochastic Petri nets*. Wiley.

Molloy M.K., 1982. *Performance Analysis Using Stochastic Petri Nets. IEEE Trans on Comput*, 31, no. 9, 913–917.

Sereno M. and Balbo G., 1997. *Mean Value Analysis of stochastic Petri nets. Perf Eval*, 29, no. 1, 35–62.

Torsello A.; Rodolà E.; and Albarelli A., 2011. *Sampling relevant points for surface registration*. In *Proc. of 3DIMPVT 2011*. 290–295.

Welch P.D., 1983. *The statistical analysis of simulation results*. The Computer Performance Modeling Handbook. Academic Press, New York.

Zimmermann A.; Freiheit J.; German R.; and Hommel G., 2000. *Petri Net Modelling and Performability Evaluation with TimeNET 3.0*. In *TOOLS '00: Proc. of the 11th Int. Conf. on Computer Performance Evaluation: Modelling Techniques and Tools*. Springer-Verlag, London, UK. ISBN 3-540-67260-5, 188–202.