URL: http://www.elsevier.nl/locate/entcs/volume52.html 18 pages

Petri Nets with Discrete Phase Type Timing: A Bridge Between Stochastic and Functional Analysis

Andrea Bobbio [1]

DISTA, Università del Piemonte Orientale, Alessandria, Italy

András Horváth [2]

Dipartimento di Informatica, Università di Torino, Torino, Italy

Abstract

The addition of timing specification in Petri Nets (PN) has followed two main lines: intervals for functional analysis or stochastic durations for performance and dependability analysis. The present paper proposes a novel technique to analyze time or stochastic PN models based on discretization. This technique can be seen as a bridge between the world of functional analysis and the world of stochastic analysis. The proposed discretization technique is based on the definition of a new construct called *Discrete Phase Type Timing - DPT* that can represent a discrete cumulative density function (cdf) over a finite support (or a deterministic cdf) as well as an interval with non-deterministic choice (or a deterministic duration). In both views, a preemption policy can be assigned and a strong (the transition must fire when the interval expires) or a weak (the transition can fire when the interval expires) firing semantics. The paper introduces the DPT construct and shows how the expanded state space can be built up resorting to a compositional approach based on Kronecker algebra. With this technique a functional model can be quantified by adding probability measures over the firing intervals without modifying the (compositional) structure of the PN model.

1 Introduction

The inclusion of timing specification in Petri net models has followed two main streams of research. In the first one time is assigned as a deterministic value (constant or interval) while in the second stream, the activities are assumed to have a random duration. We will refer to the first class of models as Time Petri Nets (TPN) and to the second class as Stochastic Petri Nets (SPN).

[1] Email: bobbio@di.unito.it
[2] Email: horvath@di.unito.it

TPN are devoted to specify and verify properties of systems where timing is a critical parameter that may affect the behavior of the system. In this line of research [18], time is assigned as a constant value or as an interval defined by a min (*earliest firing time - EFT*) and a max (*latest firing time - LFT*) value. The firing semantics is interleaving and with non-determinism (no weight is assigned to the action of atomic firing inside the allowed interval or for resolving conflicts). Further developments along this line are documented in [4,9]. In [15] a modified firing semantics is introduced: time is assigned as intervals, and firing may be forced when the maximum time expires *(strong firing semantics)* or firing may be not mandatory when the maximum time expires *(weak firing semantics)*. Analysis of TPN models involves the search for reachable conditions through the exploration of *firing zones* [4,22].

Since the initial work in [20,19], SPN have found a sound theoretical base and consolidated applications when the firing time assigned to timed transitions is an exponentially distributed random variable, so that the evolution of the system throughout its reachability graph is mapped into a continuous time Markov chain (CTMC). A number of tools exploit this paradigm and the most extensive applications are in the area of performance and dependability modeling and analysis [2]. However, reality is not always exponential and attempts have been made to include in SPN generally distributed transitions [1,7]. Particular emphasis has been devoted to models in which deterministic times [3,17] are combined with exponential random variables. In order to completely specify the non-Markovian stochastic process underlying the behavior of a SPN with generally distributed transition times, each transition is assigned an *age variable*. The way in which the age variable accounts for the time in which the transition has been enabled is governed by three *memory policies* [7]. In the *preemptive repeat different (prd)* policy (also called *enabling memory*) the age variable is reset each time the transition is disabled or fires; in the *preemptive repeat identical (pri)* policy [6], when the transition is disabled its age variable is reset, but when the transition is enabled again an identical firing time must be completed. Finally, in the *preemptive resume (prs)* policy the age variable maintains its value when the transition is disabled and then re-enabled, and is reset only when the transition fires.

Under the restriction that the marking process arising from these SPN is a Markov regenerative process [11,7], an analytical solution can be envisaged. Otherwise, an approximate solution can be obtained through a "Markovianization", by assigning to each transition a continuous Phase type distributions [8].

More recently, a new class of SPN has been explored, namely the one obtained by assigning to each timed transition a Discrete phase type (DPH) distribution [12,23,21,16]. DPH distributions are distributions arising from the time to absorption in discrete-time Markov chains with absorbing states and have been extensively explored in [5], where a fitting algorithm has been also provided. The peculiarity of the class of DPH distributions, is that it contains cdf with finite support like the deterministic or the (discrete) uniform. The use of DPH in Petri net models, allows to include cdf with finite support and any mixture of preemption policies. Moreover the transition matrix over the expanded state space may be expressed in a compositional way by means of Kronecker algebra [21], without the need of generating and storing the complete matrix. Hence, the cost of storing the model is of the same order as the cost of storing the reachability graph of the untimed PN and the solution may exploit efficient algorithms [10] for block matrices in Kronecker form.

This paper shows that the discretization technique, up to now adopted in SPN, can be seen as a bridge between the world of the functional analysis and the world of the stochastic analysis. To this end, we define an extended construct, called *Discrete Phase Type Timing (DPT)*, that encompasses the features of the DPH distributions and of the intervals (or constants) with non-determinism. By assigning to each timed transition of a PN a DPT we can build up both a functional model, in the line of those discussed in [18,4,15] and a stochastic model in the line of those discussed in [21,16]. One goal of this paper is to show that a functional model can be quantified by adding probability measures over the firing intervals without modifying the structure of the underlying PN model. Since the DPT class inherits the properties of DPH random variables and non-deterministic intervals, the DPT-PN model shares the same characteristics examined in [21,16] for DPH stochastic PN. In particular, the compositional structure of the expanded state space [21] can be exploited also for the functional model. Moreover, we can associate to any DPT a *preemption* policy, so that functional analysis can be carried out taking into account interruption and restart mechanisms that were not covered by previous models, and, furthermore, we can accommodate in the model both weak and strong firing semantics as defined in [15].

Discretization implies a state expansion and incurs in the state space explosion problems. The compositional approach via Kronecker algebra may alleviate this problem, and we can benefit from efficient storage techniques as those presented in [13]. Moreover, where DPT are used for functional analysis, the PN model may be interfaced with efficient discrete model-checking tools [14].

The paper is organized as follows. Section 2 introduces the Discrete Phase Type Timing structures that will be used to describe the local evolution of the transitions of the model. Section 3 describes the global evolution of the process and discusses the complexity and validity issues of the proposed technique. In Section 4 two demonstrative examples are given. Conclusions are drawn in Section 5.

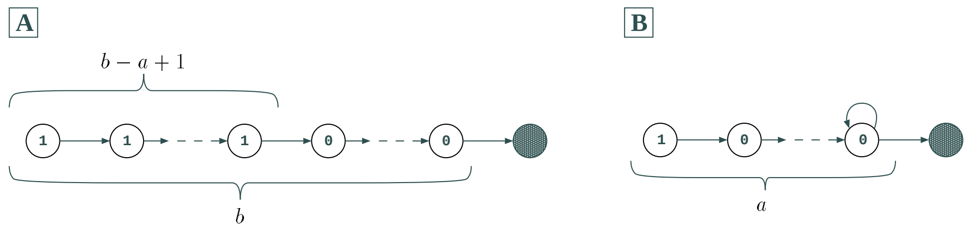

Fig. 1. **A:** Local evolution of a transition with interval firing $[a, b]$ in case of strong time semantics and prd or prs policy. **B:** Local evolution of a transition with interval firing $[a, \infty]$ in case of either weak or strong time semantics and prd or prs policy

2 Structures and matrices to describe local evolution of transitions

This section is organized as follows. Section 2.1 and Section 2.2 introduces the DPT structures and corresponding matrices for functional and stochastic analysis, respectively. These structures describe how the local descriptor of a transition evolves in a step if the transition is enabled. The applied structures depend on the adopted memory policy as well.

Throughout the paper we assume that minimal and maximal firing times are integer values and the minimal firing time is strictly positive. Note that a model in which all minimal and maximal firing times are integer multiples of a common time unit can be handled the same way. Zero minimal firing time can be handled as well by properly supplementing the model by immediate transitions.

2.1 Functional analysis

Transitions with prd or prs preemption policy

The structure used to represent the local evolution of an enabled transition with strong time semantics and firing interval $[a, b]$ is depicted in Figure 1A. When the initial phase of the structure is chosen the process may enter any of the phases signed with 1. The arrows represent the possible state-jumps; having more than one outgoing arc from a phase indicates a non-deterministic choice. The transition fires if a state-jump to the filled state occurs. In every step, if the transition is enabled, the process steps to the next phase. This structure ensures that the firing time of the transition will be in the interval $[a, b]$ and the transition fires for certain when it reaches the upper limit of its firing interval. The structure is represented by the row vector $\boldsymbol{t_0}$ that describes the possible initial phases, the square matrix $\boldsymbol{T}$ that describes the possible state-jumps and the column vector $\boldsymbol{t_f}$ that gives the phases of which firing may happen. These vectors and matrices, which will describe the local evolution of an enabled transition, are

$$\boldsymbol{t_0} = [\underbrace{1,\ldots,1}_{b-a+1},\underbrace{0,\ldots,0}_{a-1}], \quad \boldsymbol{T} = \begin{bmatrix} 0 & 1 & 0 & \ldots & \\ 0 & 0 & 1 & 0 & \ldots \\ & & \ddots & & \\ 0 & & \ldots & 0 & 1 \\ 0 & & \ldots & & 0 \end{bmatrix}, \quad \boldsymbol{t_f} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}.$$

In case of firing interval $[a, \infty]$ (still with strong time semantics) the structure depicted in Figure 1B is applied. Having been enabled for a time units the transition may either fire or remain in the last phase. Observing the structure one can easily write its descriptors $\boldsymbol{t_0}, \boldsymbol{T}$ and $\boldsymbol{t_f}$.

Assuming weak time semantics, the local evolution of an enabled transition with firing interval $[a, b]$ is followed using the structure shown in Figure 2A. When the transition is enabled for b time units it either fires or the process steps to the phase signed by the circle with thicker line from which there is no outgoing arc. This structure guarantees that the firing time will be in the interval $[a, b]$ and makes it possible that the transition does not fire in the interval. If the firing interval is $[a, \infty]$, there is no difference between strong and weak time semantics. Hence the structure depicted in Figure 1B is used.

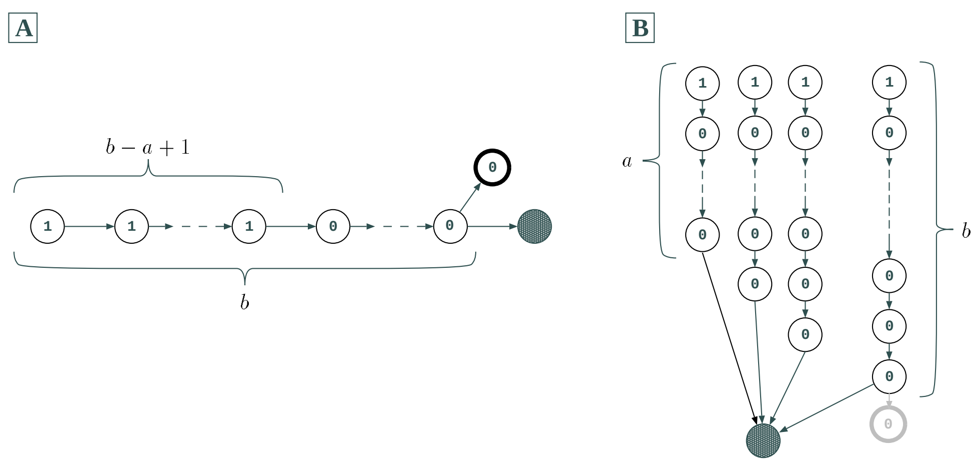

Fig. 2. **A**: Local evolution of a transition with interval firing $[a, b]$ in case of weak time semantics and prd or prs policy. **B:** Local evolution of a transition with interval firing $[a, b]$ in case of pri policy.

The above described structures are used in a different manner in case of prd than in case of prs transitions. When a prs transition is preempted the phase in which it was preempted is recorded; in case of re-enabling the process enters this state. Instead, when a prd transition is re-enabled, the initial state is chosen according to $\boldsymbol{t_0}$.

Transitions with pri policy

In case of pri transitions, the amount of time for which the transition was enabled is lost and we have to ensure that the firing time of the transition is the same after it is re-enabled. This requirement can not be fulfilled with the structures presented above. Instead, the structure of Figure 2B is used. When the transition gets preempted the column in which the process was when the preemption happened is recorded; in case of re-enabling the process enters the first phase of this column. The realization of a transition with firing interval $[a, \infty]$ would require an infinite state structure, and hence it is not considered.

When we adopt weak time semantics the gray phase drawn with thicker line is present as well. So that, it is possible that the transition does not fire in the interval $[a, b]$. When the process is in this phase when getting preempted, it returns to this phase in case of re-enabling.

2.2 Stochastic analysis

Transitions with prd or prs preemption policy

A discrete Phase type (DPH) distribution is the distribution of the time to absorption in a discrete-time Markov chain. As a counterpart of the weak time semantics in stochastic analysis, we introduce and make use of possibly defective discrete Phase type (PDDPH) distributions (see Appendix A for the definition of PDDPH distributions). A PDDPH distribution is associated to every timed transition of the Petri net. A Markov chain describing a PDDPH distribution may have two absorbing states. A jump to the absorbing phase referred to as *firing* absorbing phase (drawn with a filled circle) causes the transition to fire. A jump to the other absorbing phase, referred to as *local* absorbing phase and drawn with a thicker line, means that the transition can not fire anymore in its current sampling period (i.e. it can fire in the future only if it gets disabled and then it is re-sampled again). Without loss of generality we may assume that there are no other absorbing states or absorbing group of states in the chain.

A simple example for PDDPH distribution is depicted in Figure 3A. When the firing time of a transition is sampled the process enters one of the phases according to the initial probabilities; initial probabilities are written next to the phases. In Section 2.1 several outgoing arcs of a place represented a non-probabilistic choice. Instead, during stochastic analysis, outgoing arcs of a given phase represent a probabilistic choice among the arcs, i.e. a real value from the interval $[0, 1]$ is associated to each arc (these values are not depicted in Figure 3A) which gives the probability that a given arc will be chosen in the next step (the sum of the values associated to the outgoing arcs of a given place is 1). While in functional analysis weak time semantics implies a non-deterministic choice on whether the transition fires or not, using PDDPH distributions leads to a probabilistic choice. By computing the steady state probability of the two absorbing states one can determine how “defective” the transition is.

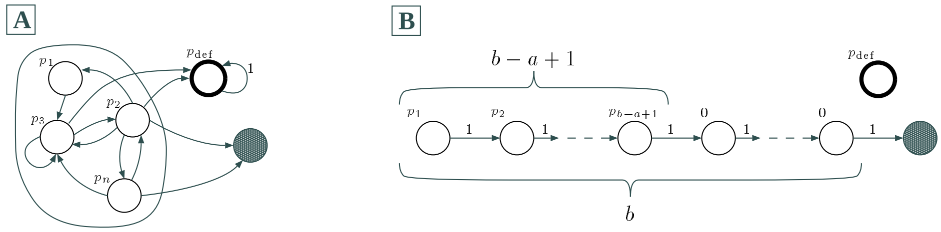

Fig. 3. **A:** Possibly defective DPH distribution. **B:** Possibly defective DPH distribution with finite support $[a, b]$ for prd or prs transitions.

A PDDPH distribution is described by its initial probability vector

$$\boldsymbol{t_0} = [p_1, p_2, \ldots, p_n, p_{\text{def}}],$$

its transition matrix $\boldsymbol{T}$ which governs the phase-jumps, and a column vector $\boldsymbol{t_f}$ which contains the probabilities of jumping to the firing absorbing state. We make two comments at this point. First, it is not necessary to have local absorbing state. Second, the row of $\boldsymbol{T}$ that corresponds to the local absorbing state contains a single non-zero value which is a 1 in the diagonal.

With PDDPH distributions one can realize a finite support distribution as shown in Figure 3B, it can be either defective or not, depending on the actual value of p_{def}. When not defective this structure is the stochastic counterpart of the structure shown in Figure 1A, when defective it is the stochastic counterpart of the structure shown in Figure 2A. In functional analysis, the only knowledge we have is that the transition fires in the interval $[a, b]$ in a non-deterministic manner. Instead, in stochastic analysis, it is possible to define the probability that the transition fires at a given time instant in the interval $[a, b]$.

Figure 4A depicts the possibility of having defective or non-defective DPH distributions with support $[a, \infty]$. The process enters either the local absorbing state or the state on the left side of the figure after which it has to take at least a steps before absorption.

The difference between handling prs and prd transitions in case of preemption is the same as when performing functional analysis.

Transitions with pri preemption policy

As for functional analysis a different structure is applied for transitions with pri policy. This structure, which is the probabilistic counterpart of the structure shown in Figure 2B, is depicted in Figure 4B.

3 Global process

For describing the global evolution of the process the method presented in [21] is followed with two important differences. First, we handle prs transitions in a manner that corresponds exactly to the definition of the prs preemption

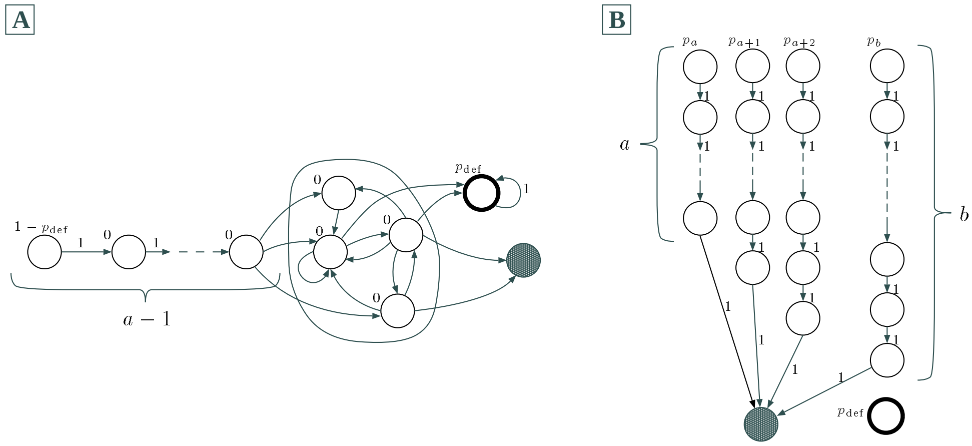

Fig. 4. **A:** Possibly defective DPH distribution with support $[a, \infty]$ for prd or prs transitions. **B:** Possibly defective DPH distribution with finite support $[a, b]$ for pri transitions.

policy [3]. Second, we extend [21] with the possibility of having transitions with pri preemption policy in the model.

In order to describe the global evolution of the process we need extended knowledge on the reachable markings of the net. We call the graph we use extended reachability graph (*erg*). A node of the *erg* carries the following information:

- the number of tokens in the places of the net,
- the set of preempted transitions with memory (the phase in which these transitions were preempted has to be recorded),
- the set of prs transitions that were candidates for firing but did not fire; these transitions, because of the definition of prs memory policy, are candidates for firing immediately when they get enabled.

The last entry of the above list requires some further explanation. In the considered model (either functional or stochastic) it can happen that two or more enabled transitions have the same firing time instant. These transitions are called candidates for firing. Having a set of candidates the resulting marking depends on the order of firings and it can happen that a transition firing prevents another candidate from firing. The set of possible orders (in case of functional analysis) or the probability of a given order (in case of probabilistic analysis) can be determined based on priority considerations; this issue is not in the scope of this paper. Not having additional priority information on the transitions, one can assume that all orders of the candidates are possible (functional analysis) or all orders have equal probability (probabilistic analysis) [21].

A simplest example of *erg* is shown in Figure 5. Even if the net has two reachable markings, still the *erg* has four nodes. In marking P_3 three

[3] A prs transition that was candidate for firing but did not fire has to be candidate for firing immediately when it gets enabled again. This fact was not considered in [21].

situations have to be distinguished: the marking was reached by the firing of T_1, the marking was reached by the firing of T_2 and T_1 was not candidate for firing, the marking was reached by the firing of T_2 and T_1 was candidate for firing.

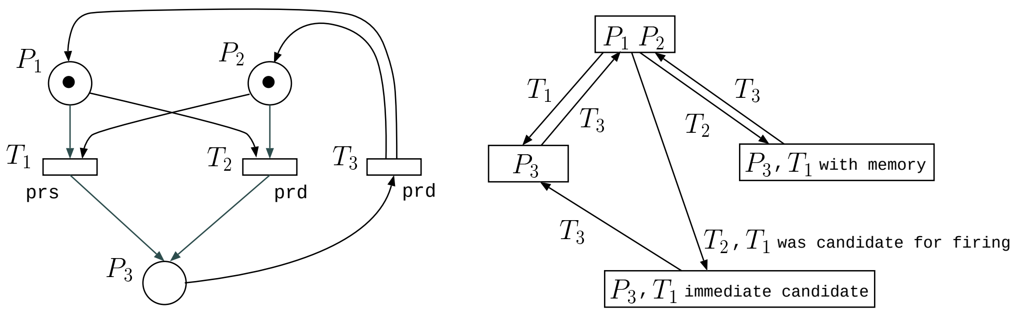

Fig. 5. In marking P_3 transition T_1 may either have or not have memory

The following notations are used to describe the procedure. We assume that the number of nodes in the *erg* is finite and denoted by N. The ith node of the *erg* will be denoted by $\boldsymbol{m}_i$. The set of prs and pri transitions are denoted by $\mathcal{S}$ and $\mathcal{I}$, respectively. The firing interval of a pri transition T_i is denoted by $[a^{(i)}, b^{(i)}]$. The set of transitions that are enabled in $\boldsymbol{m}_i$ is denoted by $\mathcal{A}_i$. The set of transitions that are disabled but have memory in $\boldsymbol{m}_i$ is denoted by $\mathcal{B}_i$. The set of transitions that were candidates but did not fire when the process entered extended marking $\boldsymbol{m}_i$ is denoted by $\mathcal{C}_i$. The number of transitions in the net is M. The transitions are ordered and the ith one is denoted by T_i.

The local evolution of a transition is described by the vectors and matrix introduced in Section 2.1 for functional and in Section 2.2 for stochastic analysis. The descriptors of transition T_i, $1 \leq i \leq M$ are denoted by $\boldsymbol{t_0}^{(i)}$, $\boldsymbol{T}^{(i)}$ and $\boldsymbol{t_f}^{(i)}$.

3.1 Global descriptor

During the analysis, the transient descriptors of the system are stored in the vectors $\boldsymbol{p}_i$, $1 \leq i \leq N$. In a vector $\boldsymbol{p}_i$ every position corresponds to a combination of local descriptors of the transitions that are enabled or are disabled but have memory in the extended marking $\boldsymbol{m}_i$. In case of functional analysis the vector $\boldsymbol{p}_i$ contains 0s and 1s. An entry 1 in $\boldsymbol{p}_i$ means that it is possible that the process is in $\boldsymbol{m}_i$ with descriptors corresponding to the position of the entry. Instead, when performing stochastic analysis, $\boldsymbol{p}_i$ may contain any real value in $[0, 1]$. In this case, an entry of $\boldsymbol{p}_i$ gives the probability that the process is in $\boldsymbol{m}_i$ with descriptors corresponding to the position of the entry.

Let us denote the number of phases of the structure representing transition T_i by n_i and the local descriptor of T_i by l_i. The vector $[l_1, l_2, \ldots, l_M]$, which contains the descriptors of all the transitions, together with the index of extended marking defines a state of the process. If a transition T_1 is disabled and does not have memory its descriptor equals 1. When T_i is enabled in an extended marking we have $1 \le l_i \le n_i$. The phase in which a prs transition gets disabled has to be recorded; for a prs transition T_i that is disabled but has memory in $\boldsymbol{m}_j$ $1 \le l_i \le n_i$. In case of a pri transition T_i with firing interval $[a^{(i)}, b^{(i)}]$ that is disabled but has memory in $\boldsymbol{m}_j$ only the column in which it got disabled is recorded (Figure 2B and 4B), hence $1 \le l_i \le b^{(i)} - a^{(i)} + 1$.

In the following, we describe how to find a given combination of the local descriptors in the vector $\boldsymbol{p}_i$. We use a so-called mixed-based numbering scheme which is closely related to the Kronecker product operator. Let us use the notation

$$n_i^{(k)} = \prod_{j=k}^{M} s_i^{(j)}, \text{ where } s_i^{(j)} = \begin{cases} 1 & \text{if } T_j \notin \mathcal{A}_i \cup \mathcal{B}_i \\ n_j & \text{if } T_j \in \mathcal{A}_i, \\ n_j & \text{if } T_j \in \mathcal{B}_i \cap \mathcal{S}, \\ b^{(j)} - a^{(j)} + 1 & \text{if } T_j \in \mathcal{B}_i \cap \mathcal{I}, \end{cases}$$

i.e. $s_i^{(j)}$ is the number of different values the descriptor l_j may have in extended marking $\boldsymbol{m}_i$. Then, in the state space spanned by the local descriptors, the possibility or the probability that the process is in discrete marking $\boldsymbol{m}_i$ and the local descriptors are $[l_1, l_2, \ldots, l_M]$ is given by the mth entry of $\boldsymbol{p}_i$ with

$$m = (\ldots((l_1 - 1)s_i^{(2)} + (l_2 - 1))s_i^{(3)} \ldots)s_i^{(M)} + l_M - 1 = \sum_{k=1}^{M}(l_k - 1)n_i^{(k+1)},$$

where $n_i^{(M+1)} = 1$.

Let us assume that the initial extended marking is $\boldsymbol{m}_i$. Then, initially

$$\boldsymbol{p}_i = \bigotimes_{k, T_k \in \mathcal{A}_i} \boldsymbol{t_0}^{(k)},$$

where $\otimes$ denotes the Kronecker-product operator (see Appendix B for the definition of the operator).

Note that none of the disabled transitions can have memory in the initial marking. All entries of all the other transient vectors are set to 0 at the beginning of the analysis.

3.2 Global evolution

In the following we show how to build the matrix $\boldsymbol{P}$ that describes the global evolution of the process.

For functional analysis the resulting matrix is the incidence matrix of the model, i.e. the value 1 (0) in position (i, j) means that macrostate j is reachable (not reachable) in one step from macrostate i. The evolution of the process is followed by successive multiplication of the transient vector $[\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_N]$ by $\boldsymbol{P}$. This way we are given the set of possible macrostates at time $1^+, 2^+, 3^+, \ldots$, i.e. right after integer multiples of the chosen time unit.

In case of stochastic analysis a probabilistic choice is defined in the intervals assigned to the transitions with a given common step-size δ. The matrix $\boldsymbol{P}$ is the transition matrix of the underlying discrete-time Markov chain, i.e. the value in position (i, j) gives the one step probability from macrostate i to macrostate j. Once again, the evolution of the process is followed by successive multiplication of the transient vector $[\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_N]$ by $\boldsymbol{P}$. This way we are given the probability of macrostates at time $\delta, 2\delta, 3\delta, \ldots$.

The matrix $\boldsymbol{P}$ is built as an $N \times N$ block-matrix in which each block is expressed as the sum of Kronecker-products of suitable matrices. The diagonal blocks $\boldsymbol{P}_{ii},\ 1 \le i \le N$ are square matrices describing the evolution of the process inside the macrostate corresponding to $\boldsymbol{m}_i$. The off-diagonal blocks $\boldsymbol{P}_{ij},\ 1 \le i, j \le N$ describe the jumps from any state of macrostate $\boldsymbol{m}_i$ to any state of macrostate $\boldsymbol{m}_j$. First we consider the case when none of the transitions fires.

Evolution when no firing happens

When none of the transitions fires in a step the process remains in the same macrostate. Hence, this case contributes to the diagonal entries $\boldsymbol{P}_{ii},\ 1 \le i \le N$ of $\boldsymbol{P}$. A diagonal block is expressed by the following Kronecker-product

$$\boldsymbol{P}_{ii} = \bigotimes_{j=1}^{M} \boldsymbol{Q}_j \tag{1}$$

where

- $\boldsymbol{Q}_j = \boldsymbol{T}^{(j)}$ if $T_j \in \mathcal{A}_i$, i.e. T_j is enabled in $\boldsymbol{m}_i$ and its evolution is described by $\boldsymbol{T}^{(j)}$;
- $\boldsymbol{Q}_j$ is an identity matrix, its size is $n_j \times n_j$ if $T_j \in \mathcal{B}_i \cap \mathcal{S}$ and $(b^{(j)} - a^{(j)} + 1) \times (b^{(j)} - a^{(j)} + 1)$ if $T_j \in \mathcal{B}_i \cap \mathcal{I}$, i.e. the descriptor of a disabled transition having memory is kept;
- $\boldsymbol{Q}_j = 1$ if $T_j \notin \mathcal{A}_i \cup \mathcal{B}_i$, i.e. the transition does not contribute to the evolution of the macrostate and has no influence on the Kronecker-product.

Evolution in case of firings

Firing of a set of transitions is considered by the expression

$$\boldsymbol{P}_{ij} += \sum_{\mathcal{L} \in S(\mathcal{A}_i)} W_{ij}(\mathcal{L}) \bigotimes_{k=1}^{M} \boldsymbol{Q}_k(\mathcal{L}) \tag{2}$$

where the function $S(\bullet)$ gives the set of non-empty subsets of its argument, and $W_{ij}(\mathcal{L})$ has the following meaning:

- In case of functional analysis its value is 1 if from extended marking $\boldsymbol{m}_i$ having the transitions in $\mathcal{L}$ as candidates the next extended marking can be $\boldsymbol{m}_j$; it is 0 otherwise. $W_{ij}(\mathcal{L})$ can be determined by considering all possible orders of $\mathcal{L}$.
- In case of stochastic analysis the value of $W_{ij}(\mathcal{L})$ is the probability that

being in $\boldsymbol{m}_i$, having the transitions in $\mathcal{L}$ as candidates the next extended marking is $\boldsymbol{m}_j$. The value of $W_{ij}(\mathcal{L})$ may be determined by assigning a probability to all possible orders of $\mathcal{L}$. The choice when all ordering has the same probability is discussed in [21].

During the calculation of $W_{ij}(\mathcal{L})$ the set $\mathcal{C}_i$ has to be considered as well.

In (2), we use $+=$ instead of $=$ because if a sequence of firings leads back to the same extended marking (it can easily happen in the presence of immediate transitions) the quantity of the right hand side is added to the quantity given in (1). The term $\boldsymbol{Q}_k(\mathcal{L})$ is determined according to the following situations:

- If $T_k \in \mathcal{L}$, i.e. if T_k is one of the candidates the following cases have to distinguished:
  - if T_k is neither enabled nor it has memory in $\boldsymbol{m}_j$ (i.e. $T_k \notin \mathcal{A}_j \cup \mathcal{B}_j$), then $\boldsymbol{Q}_k = \boldsymbol{t_f}^{(k)}$,
  - if T_k is re-enabled in $\boldsymbol{m}_j$ (i.e. $T_k \in \mathcal{A}_j$), then $\boldsymbol{Q}_k = \boldsymbol{t_f}^{(k)}\boldsymbol{t_0}^{(k)}$,
  - if T_j is not enabled but has memory in $\boldsymbol{m}_j$ (i.e. $T_k \in \mathcal{B}_j$) which can happen as a result of a sequence of firing, then $\boldsymbol{Q}_k = \boldsymbol{t_f}^{(k)}\boldsymbol{t_0}^{(k)}$.
- If $T_k \in (\mathcal{A}_i/\mathcal{L}) \cap (\mathcal{A}_j \cup \mathcal{B}_j)$, i.e T_k is enabled in $\boldsymbol{m}_i$, it does not fire and is enabled or has memory in $\boldsymbol{m}_j$, then $\boldsymbol{Q}_k = \boldsymbol{T}^{(k)}$.
- If $T_k \in (\mathcal{A}_i/\mathcal{L})$ and $T_k \notin \mathcal{A}_j \cup \mathcal{B}_j$, i.e. T_k gets disabled and does not have memory in $\boldsymbol{m}_j$, then $\boldsymbol{Q}_k = \boldsymbol{e}^{(k)} - \boldsymbol{t_f}^{(k)}$ (where $\boldsymbol{e}^{(k)}$ is a vector of size n_k with all entries equal to one).
- If $T_k \in (\mathcal{B}_i \cap \mathcal{B}_j)$, i.e. it is not enabled but has memory in both $\boldsymbol{m}_i$ and $\boldsymbol{m}_j$, then $\boldsymbol{Q}_k$ is the identity matrix of proper size (its size is $n_k \times n_k$ if T_k is of prs size, while its size is $b^{(k)} - a^{(k)} + 1 \times b^{(k)} - a^{(k)} + 1$ if T_k is of pri type).
- If T_k is of prs type and $T_k \in (\mathcal{B}_i \cap \mathcal{A}_j)$, i.e. it is not enabled but has memory in $\boldsymbol{m}_i$ and gets enabled in $\boldsymbol{m}_j$, then $\boldsymbol{Q}_k$ is the identity matrix of size $n_k \times n_k$.
- If T_k is of pri type and $T_k \in (\mathcal{B}_i \cap \mathcal{A}_j)$, i.e. it is not enabled but has memory in $\boldsymbol{m}_i$ and gets enabled in $\boldsymbol{m}_j$, then $\boldsymbol{Q}_k$ is of size $b^{(k)} - a^{(k)} + 1 \times n_k$ and defined as

$$[\boldsymbol{Q}_k]_{ij} = \begin{cases} 1, \text{ if } j = (i-1)a^{(k)} + i, \\ 0, \text{ otherwise.} \end{cases}$$

  This matrix insures that being re-enabled the local descriptor of the pri transition corresponds to the first phase of the proper column.
- If $T_k \notin (\mathcal{A}_i \cap \mathcal{B}_i)$ and $T_k \in (\mathcal{A}_j \cap \mathcal{B}_j)$, i.e. it is neither enabled nor has memory in $\boldsymbol{m}_i$ and either enabled or has memory in $\boldsymbol{m}_j$, then $\boldsymbol{Q}_k = \boldsymbol{t_0}^{(k)}$.
- If T_k is neither enabled nor has memory in both in $\boldsymbol{m}_i$ and $\boldsymbol{m}_j$, then $\boldsymbol{Q}_k = 1$, and hence does not have any influence on the Kronecker-product.

3.3 Complexity

The complexity of the proposed procedure is different for functional than for stochastic analysis for two reasons.

- The amuont of memory needed to store the structure that describes the local evolution of the process is usually larger in case of stochastic analysis. In case of functional analysis, a transition with firing interval [1 : 2] is represented by a two phase structure. In case of probabilistic analysis a discrete probability mass function is placed in the interval. If it is done with step-size 0.2 (which means that the firing time may be 1,1.2,1.4,...,2.0) 10 phases are needed.
- In case of functional analysis, a state described by a global descriptor is possible or not in a given time instant. So that the transient vectors contain 0s and 1s which can be stored efficiently using binary decision diagrams. In case of probabilistic analysis the transient vectors contain real values whose storage requires more memory.

The length of the transient vector describing an extended marking is the product of the number of different values that the local descriptors may have in that marking. This product can be very large if many transitions are enabled or have memory in an extended marking.

The matrix describing the global evolution of the process is stored in a memory-efficient manner by using Kronecker-expressions. Different algorithms can be implemented to perform Kronecker-products, a comparison of the complexity of these algorithm is found in [10].

3.4 Validity of the proposed approach

In case of stochastic analysis, the proposed technique describes correctly the evolution of the model. In case of functional analysis, two cases have to distinguished:

- If the time of the model is discrete, i.e. duration of actions may be only integer multiples of the time unit, the approach results in correct tracking of the behavior of the model.
- Instead, if we assume dense time, i.e. durations may take any real value from the firing intervals of the transitions, considering only integer multiples of the time unit (as we do in the proposed approach) possible execution sequences might be excluded. In fact, on the one hand, it is possible to prove that the technique is correct if the model contains only prd transitions. On the other hand, if prs or pri transitions are present, some execution sequences can be missed out, and, as a consequence, the presented technique results only in an approximation of the behavior of the model.

4 Examples

Two examples are given in this section. The first, simple one is presented to show that a net may have different functional behaviors for different preemption policies. On the second example both functional and stochastic analysis are performed.

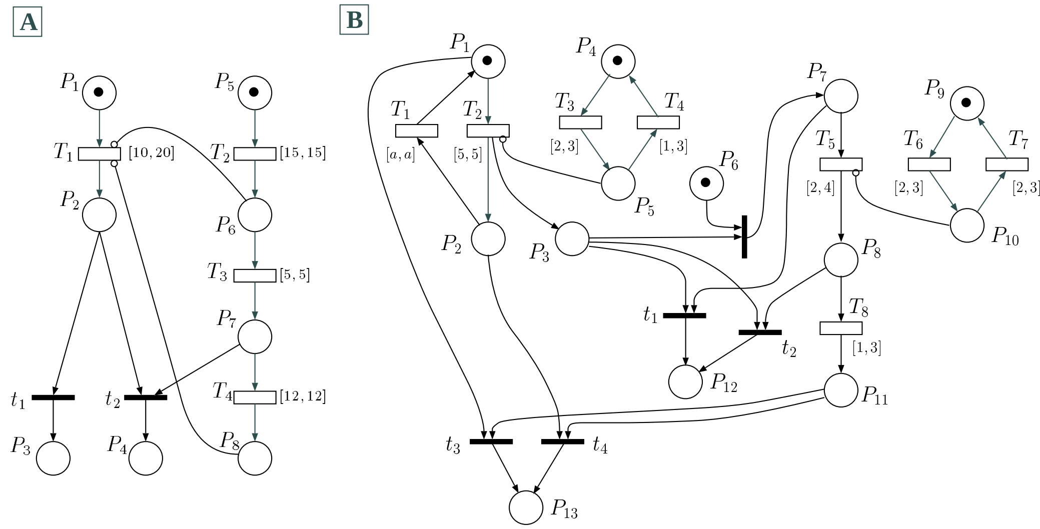

Fig. 6. **A:** Different reachable markings in case of adopting different preemption policies for T_1. **B:** Producer-consumer model.

The time Petri net depicted in Figure 6A has different reachable markings when adopting different preemption policies for transition T_1. We assume strong time semantics. Whichever policy is chosen, it is possible that T_1 fires before T_2 (if the firing time of T_1 is chosen lower than 15); in this case the net ends up in marking (P_3, P_8). If T_2 fires before T_1 the behavior depends on the memory policy of T_1:

- prd policy: T_1 either fires or does not fire in marking (P_1, P_7) (it fires if its resampled firing time is less than 12),
- prs policy: T_1 fires in marking (P_1, P_7) since its "clock" starts from 15 entering this marking and its maximal firing time is 20,
- pri policy: T_1 does not fire in marking (P_1, P_7) since its firing time is not resampled and it is greater than 15.

The second example, which is a simple producer-consumer model, is depicted in Figure 6B. Production, represented by deterministic transition T_2, may get preempted. The preemption is modeled by transitions T_3 and T_4, and places P_4 and P_5. Transition T_2 is of prs type, i.e. the work done is not lost in case of preemption. Production restarts after a time units (represented by transition T_1). Consumption consists of two steps represented by transitions T_5 and T_8. The first phase of consumption may get preempted which feature is modeled by the subnet P_9, P_{10}, T_6 and T_7. For transition T_5 prd or prs memory policy is considered. The aim of the analysis is to determine if the consumer finishes its two-phase job before the arrival of the next one. A token in place P_{12} indicates an error (i.e. another job arrived before the consumer finished the previous one), while a token in place P_{13} indicates that one cycle of production-consumption was successful. The model is evaluated with strong time semantics.

From the functional point of view possible questions are: "Is it possi-

Analysis	prd, $a=8$	prd, $a=20$	prs, $a=8$	prs, $a=20$
Functional	2133	3645	3249	5625
Stochastic	14696100	24236100	16964100	46052100

Table 1
Size of the discretized state space

ble that a token appears in place P_{12}?" or "What is the shortest/longest cycle-time?". From the stochastic point of view one could ask: "What is the probability that a token appears in place P_{12}?" or "What is the probability of successfully finishing a cycle before time t?". In case of performing stochastic analysis a discrete probability distribution is defined on the interval. For all non-deterministic transitions we assumed to have discrete uniform distribution with step-size 0.1 (for example the firing time of transition T_3 may be 2.0,2.1,2.2,...,2.9 or 3.0 with equal probability of 1/11).

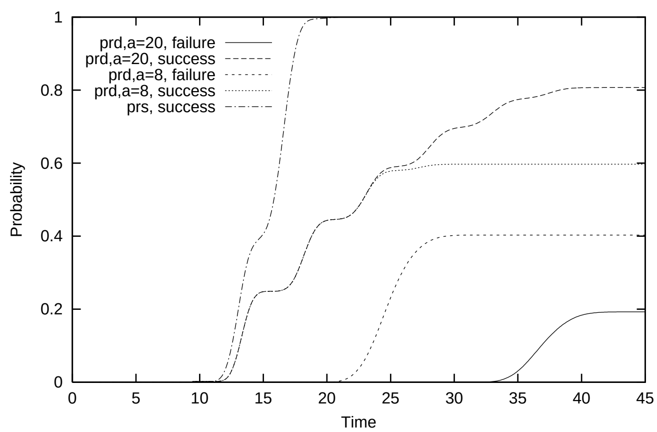

Fig. 7. Probability of failure or correctly finished cycle as a function of time

The example was evaluated, using a preliminary implementation of the presented Kronecker-based description, with $a=8$, $a=20$ and with either prd or prs memory policy for transition T_5. From the point of view of functional analysis the results are the following. For both values of a: if the adopted memory policy is prd consumption may either terminate in time or may not, while for prs policy it terminates always before the next production. The earliest possible time to finish correctly the production and consumption is 9 time units for all the cases. As a probabilistic result, Figure 7 depicts the probability of having a correct or erroneous outcome as a function of time (in case of prs policy for transition T_5, erroneous outcome is not possible and for both values of a we have the same curve).

The size of the discretized state space for the different cases is given in Table 1. As one could observe functional analysis requires much smaller state space.

5 Conclusion

The paper has introduced a new construct called *Discrete Phase Type Timing - DPT* that can represent probabilistic or non-deterministic choice over an interval. For both cases it gives the possibility of assigning preemption policy to the transitions of the system. Both weak and strong time semantics can be handled (or even mixed in the same model).

A compositional approach, based on Kronecker algebra, was given to build the matrix that describes the evolution of the expanded state space. This description is similar to the one given in [21] with the differences that it follows the behavior of a prs transition in an exact manner and provides the possibility of having pri transitions in the model. It was shown as well that the same compositional description can be utilized for functional and for stochastic analysis.

Through a simple example the possibility of performing both functional and probabilistic analysis of the same model has been demonstrated.

References

[1] M. Ajmone Marsan, G. Balbo, A. Bobbio, G. Chiola, G. Conte, and A. Cumani. The effect of execution policies on the semantics and analysis of stochastic Petri nets. *IEEE Transactions on Software Engineering*, 15:832–846, 1989.

[2] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis. *Modelling with Generalized Stochastic Petri Nets*. Wiley Series in Parallel Computing, 1995.

[3] M. Ajmone Marsan and G. Chiola. On Petri nets with deterministic and exponentially distributed firing times. In *Lecture Notes in Computer Science*, volume 266, pages 132–145. Springer Verlag, 1987.

[4] B. Berthomieu and M. Diaz. Modelling and verification of time dependent systems using Time Petri Nets. *IEEE Transactions on Software Engineering*, 17(3):259–273, 1991.

[5] A. Bobbio, A. Horváth, M. Scarpa, and M. Telek. Acyclic discrete phase type distributions: Properties and a parameter estimation algorithm. Technical Report of Budapest University of Technology and Economics - Submitted for publication, 2000.

[6] A. Bobbio, V.G. Kulkarni, A. Puliafito, M. Telek, and K. Trivedi. Preemptive repeat identical transitions in Markov Regenerative Stochastic Petri Nets. In *Petri Nets and Performance Models '95*, pages 113–122. IEEE CS Press, 1995.

[7] A. Bobbio, A. Puliafito, and M. Telek. A modeling framework to implement combined preemption policies in MRSPNs. *IEEE Transactions on Software Engineering*, 26:36–54, 2000.

[8] A. Bobbio and M. Telek. Non-exponential stochastic Petri nets: an overview of methods and techniques. *Computer Systems: Science & Engineering*, 13(6):339–351, 1998.

[9] G. Bucci and E. Vicario. Compositional validation of time-critical systems using communicating time Petri nets. *IEEE Transactions on Software Engineering*, 21:969–992, 1995.

[10] P. Buchholz, G. Ciardo, P. Kemper, and S. Donatelli. Complexity of memory-efficient Kronecker operations with applications to the solution of Markov models. *INFORMS Journal on Computing*, 13(3):203–222, 2000.

[11] Hoon Choi, V.G. Kulkarni, and K. Trivedi. Markov regenerative stochastic Petri nets. *Performance Evaluation*, 20:337–357, 1994.

[12] G. Ciardo. Discrete-time markovian stochastic petri nets. In W. J. Stewart, editor, *Computations with Markov Chains*, pages 339–358. Kluwer, 1995.

[13] G. Ciardo and A. Miner. A data structure for the efficient Kronecker solution of GSPNs. In *Proc. 8th Int. Workshop on Petri Nets and Performance Models (PNPM'99)*, pages 22–31. IEEE CS Press, 1999.

[14] E. M. Clarke, E. A. Emerson, and A. P. Sistla. Automatic verification of finite state concurrent systems using temporal logic specifications: A practical approach. *ACM Transactions on Programming Languages and Systems*, 8(2):244–263, 1986.

[15] C. Ghezzi, D. Mandrioli, S. Morasca, and M. Pezzeè. A unified high level Petri net formalism for time-critical systems. *IEEE Transactions on Software Engineering*, 17:160–171, 1991.

[16] A. Horváth, A. Puliafito, M. Scarpa, and M. Telek. A discrete time approach to the analysis of non-Markovian stochastic Petri nets. In *Tools 2000*, volume 1786 of *Lecture Notes in Computer Science*, pages 171–187, Schaumburg, IL, USA, March 2000. Springer-Verlag.

[17] C. Lindemann. *Performance Modelling with Deterministic and Stochastic Petri Nets*. John Wiley, 1998.

[18] P. Merlin and D. J. Faber. Recoverability of communication protocols. *IEEE Transactions on Communication*, 24(9):1036–1043, 1976.

[19] M.K. Molloy. On the integration of delay and throughput measures in distributed processing models. Technical report, Phd Thesis, UCLA, 1981.

[20] S. Natkin. Les reseaux de Petri stochastiques et leur application a l'evaluation des systemes informatiques. Technical report, These de Docteur Ingegneur, CNAM, Paris, 1980.

[21] M. Scarpa and A. Bobbio. Kronecker representation of Stochastic Petri nets with discrete PH distributions. In *International Computer Performance and Dependability Symposium - IPDS98*, pages 52–61. IEEE CS Press, 1998.

[22] E. Vicario. Static analysis and dynamic steering of time dependent systems. *IEEE Transactions on Software Engineering (to be published)*, 2001.

[23] R. Zijal, G. Ciardo, and G. Hommel. Discrete deterministic and stochastic petri nets. In *Proc. Measurement, Modeling, and Valuation of Computer- and Communication-Systems (MMB)*, pages 103–117, Freiberg, Germany, 1997. VDE-Verlag.

Appendix A: Discrete phase type distributions

Possibly defective discrete phase type (PDDPH) distributions are defined in terms of discrete-time Markov chains (DTMC) with absorbing states. A discrete random variable X is PDDPH distributed if and only if there exists a DTMC $\{Z_i, i \geq 0\}$ with $n+1$ states of which the $(n+1)^{th}$ is absorbing (there can be other absorbing states as well) and $Pr\{X \leq i\} = Pr\{Z_i = n+1\}$, i.e. X is the time to reach state $n+1$. If state $n+1$ is not the only absorbing state the distribution can be defective. A PDDPH distribution is given by the initial probability vector of its DTMC ($\boldsymbol{t_0}$) and the one-step transition matrix ($\boldsymbol{T}$) that governs the evolution among the states excluding state $n+1$. Then the distribution of X is given by $Pr\{X \leq i\} = 1 - \boldsymbol{t_0}\boldsymbol{T}^i\boldsymbol{e}$, where $\boldsymbol{e}$ is a vector with all entries equal to 1.

Appendix B: Kronecker-product operator

The Kronecker-product $\boldsymbol{C} = \boldsymbol{A} \otimes \boldsymbol{B}$ of matrix $\boldsymbol{A}$ of size $(a_r \times a_c)$ and matrix $\boldsymbol{B}$ of size $(b_r \times b_c)$ is of size $(a_r b_r \times a_c b_c)$ and is defined by

$$\boldsymbol{C}_{i,j} = \boldsymbol{A}_{i_2,j_2}\boldsymbol{B}_{i_1,j_1}, \text{ where } i = i_2 a_r + i_1, j = j_2 a_c + j_1.$$