©2009 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

State-Density Functions over DBM Domains in the Analysis of Non-Markovian Models

Laura Carnevali, Leonardo Grassi, and Enrico Vicario, *Member*, *IEEE*

Abstract—Quantitative evaluation of models with generally distributed transitions requires the analysis of non-Markovian processes that may be not isomorphic to their underlying untimed models and may include any number of concurrent nonexponential timers. The analysis of stochastic Time Petri Nets (sTPNs) copes with the problem by covering the state space with stochastic classes, which extend the theory of Difference Bounds Matrix (DBM) with a state probability density function. As a core step, the analysis process requires symbolic manipulation of density functions supported over DBM domains. We characterize and engineer the critical steps of this derivation. We first show that the state-density function accepts a continuous piecewise representation over a partition in DBM-shaped subdomains. We then develop a closed-form symbolic calculus of state-density functions under the assumption that transitions in the sTPN model have expolynomial distributions over possibly bounded intervals. The calculus shows that within each subdomain, the state-density function is a multivariate expolynomial function, and it makes explicit the way in which this form evolves and grows in complexity as the state accumulates memory through subsequent transitions. This enables an efficient implementation of the analysis process and provides the formal basis that supports the introduction of an imprecise analysis based on the approximation of state-density functions through Bernstein Polynomials. The approximation attacks practical and theoretical limits in the applicability of stochastic state classes and devises a new approach to the analysis of non-Markovian models, relying on approximations in the state space rather than in the structure of the model.

Index Terms—Correctness verification, performance and dependability, quantitative evaluation, stochastic Time Petri nets, dense-time state-space analysis, Difference Bounds Matrix, Markov Renewal Theory, approximate state-space representation, density function approximation, Bernstein polynomials.

1 Introduction

Qualitative verification and quantitative evaluation play complementary roles in the engineering of concurrent and timed software. While the former identifies feasible behaviors, the latter associates them with a measure of probability. This assumes a crucial relevance in various processes of the development lifecycle of safety critical software components. Unfortunately, the two aspects of verification and evaluation determine contrasting needs, resulting in separate formalisms and techniques.

In qualitative verification, the focus on the identification of feasible behaviors places a major relevance on the representation of concurrent nondeterministic timers taking values within a possibly bounded support. This is essential to represent implicit precedences induced by the relative timing of concurrent events, and comprises a common element of the expressivity of most verification formalisms, such as Time Petri Nets and Timed Automata [28], [3]. Since timing constraints restrict the set of feasible behaviors, state-space enumeration must jointly account for logical locations and timers. And, since timers are densely valued, states cannot be enumerated explicitly. The state space must thus be covered with *state classes*, each encoding a dense variety of states with the same logical location but with different values of the vector of active timers [37]. Different abstractions can be considered to encode the domain of timers values included within a class [2], [19], [5]. A large part of the literature has developed on Difference Bounds Matrix (DBM) [19], [37], [6], which combines various fortunate properties: it is closed with respect to the transformation induced by the advancement of time observed at the firing of a transition; it provides a compact partition of the state space such that any two states reached through the same sequence of logical locations are included in the same class; it accepts compact encoding and efficient manipulation algorithms. These properties have been exploited in a variety of verification techniques and tools [4], [12], [21], [8], [18], which solve crucial decision and optimization problems pertaining timed reachability and maximum duration between events. However, while supporting the identification of feasible behaviors, these techniques do not provide any measure of their probability.

In quantitative evaluation, nondeterministic choices and timers are associated with a probability measure so that the evolution of the logical location becomes a continuous time stochastic chain. In this perspective, models for quantitative evaluation can be regarded as an extension of those for qualitative verification. However, in order to enable Markovian analysis techniques, the representation of concurrent timers is subject to severe restrictions pertaining the type of distributions, their support, and/or the degree of concurrency among active timers. If all timers are distributed as negative exponential variables over an unbounded support, as in Stochastic Petri Nets (SPNs) [29],

• *The authors are with the Dipartimento di Sistemi e Informatica—Università di Firenze 3, via S. Marta, 50139, Firenze, Italy.*
E-mail: {carnevali, grassi, vicario}@dsi.unifi.it.

Manuscript received 27 Jan. 2008; revised 1 July 2008; accepted 8 Aug. 2008; published online 2 Dec. 2008.
Recommended for acceptance by J. Hillston, M. Kwiatkowska, and M. Telek. For information on obtaining reprints of this article, please send e-mail to: tse@computer.org, and reference IEEECS Log Number TSESI-2008-01-0048. Digital Object Identifier no. 10.1109/TSE.2008.101.

0098-5589/09/$25.00 © 2009 IEEE Published by the IEEE Computer Society

Authorized licensed use limited to: Universita degli Studi di Firenze. Downloaded on July 2, 2009 at 03:23 from IEEE Xplore. Restrictions apply.

the stochastic process underlying the model is a Continuous Time Markov Chain (CTMC) isomorphic to the state space of the untimed model, which can be efficiently analyzed in transient and steady state [17]. If immediate transitions are added, as in Generalized Stochastic Petri Nets (GSPNs) [1], the reduced process that observes only tangible states is still a CTMC isomorphic to a state space that can be derived from the untimed model.

When the model includes generally distributed timers, the underlying process becomes a generalized semi-Markov process [16] for which no general techniques of analysis are available. As an additional difficulty, if some of the timers take values over a bounded support, the space of reachable markings is not isomorphic to that of the untimed model. Both complexities can be avoided in various special cases. If the model never reaches a state where two generally distributed transitions are persistent, then the underlying process becomes a semiregenerative Markov process and can be analyzed through Markov Renewal Theory [16], [15] or through the method of supplementary variables [22], [23], [35]. In [24], [25], the limit on the number of concurrent generally distributed transitions is broken for the special case of deterministic transitions through the enumeration of a discrete-time general state-space Markov Chain (GSSMC).

Following a different approach, various works have proposed approximation of generally distributed transitions to obtain an approximant model that underlies a CTMC. In [9], generally distributed transitions are approximated through continuous Phase Type distributions [30]. Approximation based on the subclasses of Coxian distributions and Exponential Mixtures are proposed in [27] and [20], respectively. In the perspective of this work, these approaches share two common salient traits: They apply the approximation in the structure of the model and they resort to approximant functions supported over unbounded domains. The former aspect enables usage of monovariate approximant functions. The latter guarantees isomorphism between the state space of the untimed model and the underlying stochastic process.

In [11], [13], we introduced a new technique that integrates qualitative verification and quantitative evaluation of densely timed models with any number of concurrent generally distributed timers. The method develops on the concept of *stochastic classes*, which augment state classes with a *state-density function*. This establishes a probability measure over the variety of timed behaviors encompassed within the boundaries of a state space covered through state classes. The reachability relation among stochastic classes defines a *stochastic class graph*, which enables the integration of correctness verification based on the theory of DBM classes with the derivation of various quantitative measures: the probability of any untimed firing sequence (symbolic run) and the distribution of the time spent in its execution; the identification of a set of regeneration classes and the derivation of the subordinate process between any two regeneration classes; the steady-state probabilities of regeneration classes based on Markov Renewal Theory [13].

As a core step, the analysis process requires that the enumeration of state classes [7], [37] be coupled with the symbolic manipulation of state-density functions. The approach was first implemented in the Oris tool [32] by delegating the symbolic calculus to the Mathematica Kernel [38]. However, the implementation encounters various practical and theoretical limits: Each state class may be expanded into multiple stochastic classes, exacerbating the state explosion problem of nondeterministic analysis; the expansion may be even unbounded if the state class graph includes any cycle that never visits any *regeneration* class [13]; the complexity of derivation of a successor depends not only on the structure of the state class (i.e., on the number of enabled transitions) but also on the complexity of the density function, which may grow along complex nonregenerative behaviors.

In this paper, we provide a closed-form calculus for the derivation of state-density functions over DBM domains for models where all transitions have expolynomial distributions over nonpoint-like intervals. Immediate and deterministic transitions are excluded to limit the complexity of the treatment, but they can be easily encompassed in the light of the theory developed in [13]. The derivation provides practical and conceptual results. In the practical perspective, it enables a dedicated C++ implementation, which changes the order of complexity for the time spent in the derivation and provides a library that can be embedded in a variety of analysis tools. In the conceptual perspective, the derivation makes evident the structure of density functions, the way in which they evolve through transitions between classes, and the factors that determine the actual complexity of their derivation. Based on this insight, we propose an approximate representation of the state space that leverages upon the relaxation of the equivalence between state-density functions and upon the approximation of state-density functions themselves. In particular, we develop and experiment an approximation based on Bernstein Polynomials [26], [34] that can be applied when all model timers with unbounded support have exponential distribution. This devises a new approach to the analysis of non-Markovian models, which relies on multivariate approximation performed in the state space rather than in the structure of the net. Computational experience shows that this enables the treatment of models that cannot be managed by the exact analysis for practical factors or for theoretical limits of unboundedness.

The rest of the paper is organized asfollows: In Sections 2 and 3, we recall the definition of stochastic Time Petri Nets (sTPNs) and their analysis technique based on stochastic classes. The closed-form calculus supporting symbolic derivation of state-density functions is developed in Section 4. The theory of approximation is developed in Section 5 and assessed through computational experience in Section 6. Conclusions are drawn in Section 7. For the sake of readability, all proofs and an instrumental algorithm are reported in the Appendix.

2 Stochastic Time Petri Nets

Syntax. An sTPN is a tuple $\langle P; T; A^-; A^+; m_0; EFT^s; LFT^s; \tau_0; \mathcal{F}; \mathcal{C}\rangle$. The first five members are the basic model of Petri Nets: P is a set of places; T a set of transitions disjoint from P; $A^- \subseteq P \times T$ and $A^+ \subseteq T \times P$ are pre and postconditions; $m_0 : P \to \mathbb{N}$ is the (initial) marking. EFT^s,

Authorized licensed use limited to: Universita degli Studi di Firenze. Downloaded on July 2, 2009 at 03:23 from IEEE Xplore. Restrictions apply.

LFT^s, and τ_0 add timing constraints in the style of Time Petri Nets: τ_0 associates each transition $t \in T$ with an (initial) time to fire ($\tau_0 : T \to \mathbb{R}_0^+$, where $\mathbb{R}_0^+$ denotes the set of nonnegative real numbers); besides, $EFT^s : T \to \mathbb{R}_0^+$ and $LFT^s : T \to \mathbb{R}_0^+ \cup \{+\infty\}$ associate each transition with a static firing interval made of a static Earliest Firing Time and a static Latest Firing Time, respectively, with $EFT^s(t) \leq LFT^s(t)$. Finally, $\mathcal{F}$ and $\mathcal{C}$ define a measure of probability for nondeterministic choices: $\mathcal{C}$ associates each transition with a real-valued weight ($\mathcal{C} : T \to \mathbb{R}^+$), while $\mathcal{F}$ associates each transition $t \in T$ with a static probability distribution $F_t^s()$ supported in the static firing interval $[EFT^s(t), LFT^s(t)]$.

State. The state of an sTPN is a pair $s = \langle m, \tau \rangle$, where $m : P \to \mathbb{N}$ is a marking and $\tau : T \to \mathbb{R}_0^+$ associates each transition with a time to fire.

Firability. A transition t_o is *enabled* if each of its input places contains at least one token ($\langle p, t_o \rangle \in A^- \to M(p) > 0$), and it is firable if it is enabled and its time to fire $\tau(t_o)$ is not higher than that of any other enabled transition ($\tau(t_o) \leq \tau(t_i) \ \ \forall t_i \in T^e(m)$, where $T^e(m)$ denotes the set of transitions enabled by m. When multiple transitions are firable, the choice is resolved by a random switch determined by the weight {\cal C}():

$$Prob\{t_0 \text{ is selected}\} = \frac{\mathcal{C}(t_o)}{\sum_{t_i \in T^f(s)} \mathcal{C}(t_i)},$$

where $T^f(s)$ is the set of transitions that are firable in state s. Since the treatment does not include immediate and determinate transitions, $\mathcal{C}$ turns out to be irrelevant for the purposes of the analysis [13], but it is essential in the formulation of the semantics to associate each choice with a measure of probability [36].

Firing. When a transition t_o fires, the state $s = \langle m, \tau \rangle$ is replaced by $s' = \langle m', \tau' \rangle$, which we write as $s \xrightarrow{t_o} s'$. Marking m' is derived from m by removing a token from each input place of t_o, and by adding a token to each output place of t_o

$$\begin{aligned} m_{tmp}(p) &= m(p) - 1 \quad \forall p. \langle p, t_o \rangle \in A^-, \\ m'(p) &= m_{tmp}(p) + 1 \quad \forall p. \langle t_o, p \rangle \in A^+. \end{aligned} \tag{1}$$

Transitions that are enabled both by the intermediate marking m_{tmp} and by m' are said *persistent*, while those that are enabled by m' but not by m_{tmp} are said *newly enabled*. If t_o is still enabled after its own firing, it is always regarded as newly enabled [6], [37]. For any transition t_i that is persistent after the firing of t_o, the time to fire is reduced by the time elapsed in the previous state: $\tau'(t_i) = \tau(t_i) - \tau(t_o)$. Whereas, for any transition t_a that is newly enabled after the firing of t_o, the time to fire takes a random value sampled in the static firing interval according to the static probability distribution $F_{t_a}^s()$

$$\begin{aligned} EFT^s(t_a) \leq \tau'(t_a) \leq LFT^s(t_a), \\ \text{Prob}\{\tau'(t_a) \leq x\} = F_{t_a}^s(x). \end{aligned} \tag{2}$$

3 Analysis of Stochastic Time Petri Nets

The state space of an sTPN can be covered through stochastic state classes, each including a set of states with the same marking but with different values of the vector of times to fire distributed within a DBM domain [19], [37] according to a state-density function [11], [13].

3.1 Stochastic State Classes

A *stochastic state class* (stochastic class for short) is a triple $\langle m, D, F_\tau() \rangle$, where m is a marking, τ is the vector of times to fire of transitions enabled by m, and $F_\tau()$ is a probability distribution for τ supported over *domain* D. Stochastic classes are associated with a succession relation extending the relation of AE reachability between state classes [31] with a measure of probability [11], [37]:

Definition 3.1. *Given two stochastic classes $\Sigma = \langle m, D, F_\tau() \rangle$ and $\Sigma' = \langle m', D', F_{\tau'}() \rangle$, we say that Σ' is a successor of Σ through t_o with probability μ, and we write $\Sigma \overset{t_o,\mu}{\Longrightarrow} \Sigma'$, iff the following property holds: If the marking of the net is m and the vector of times to fire of transitions enabled by m is a random variable τ with support D and distribution $F_\tau()$, then the firing of t_o occurs with probability $\mu > 0$ and leads to a new marking m' and a new vector of times to fire τ' with support D' and distribution $F_{\tau'}()$.*

Given an initial stochastic class Σ^o, the transitive closure of $\overset{t,\mu}{\Longrightarrow}$ defines a *stochastic class graph* (stochastic graph for short), where vertices are stochastic classes and edges are labeled with a transition t and a probability μ.

3.2 DBM Domains

In general, firing domains of the classes in the stochastic graph may have a different structure depending on the shape of the firing domain of the initial class. A special notable case occurs when the firing domain of the initial class is the space of solutions of a set of linear inequalities in the form of a DBM [19], [37]

$$D = \{ \tau_i - \tau_j \leq b_{ij} \quad \forall i \neq j \in [0, N-1] \cup \{*\}, \tag{3}$$

where $\tau = \langle \tau_o, \tau_1, \ldots, \tau_{N-1} \rangle$ is the vector of times to fire of enabled transitions, $\tau_* = 0$ is a fictitious variable introduced to regularize the structure of inequalities, and $b_{ij} \in \mathbb{R} \cup \{+\infty\}$ are the coefficients which define the firing domain.

DBM encoding is closed with respect to the transformation induced by the firing semantics of Time Petri Nets according to the AE succession relation among state classes: If the domain of the initial stochastic class Σ^o is encoded as a DBM, then all the classes in the stochastic graph have a domain in DBM form [19], [37]. Moreover, DBM encoding is efficient with respect to both space of representation and time of manipulation. Specifically, the DBM form has a *normal form* of representation where b_{ij} coincides with the maximum value that can be attained by the difference $\tau_i - \tau_j$ in some solution of the set. The normal form exists uniquely, supporting efficient test of equivalence in time $O(N^2)$ and detection of successor classes in time $O(N)$ [37]. The normal form is univocally identified by the condition

$$b_{ij} \leq b_{ih} + b_{hj} \quad \forall i, j, h \in [0, N-1] \cup \{*\} \text{ with } i \neq j \neq h \neq i. \tag{4}$$

According to (4), the normal form is the solution of an all-shortest-path problem and can be computed in time $O(N^3)$

Authorized licensed use limited to: Universita degli Studi di Firenze. Downloaded on July 2, 2009 at 03:23 from IEEE Xplore. Restrictions apply.

or even in time $O(N^2)$ in a repeated derivation exploiting warm restart [37].

3.3 Enumeration of Stochastic-Classes

Enumeration of the reachability relation $\overset{t,\mu}{\Longrightarrow}$ requires: detection of successors, evaluation of their probability, derivation of successor domains, and calculus of their state-density functions. We recall here the formulation provided in [13], limiting the treatment to the case of transitions with *nonpoint-like* firing intervals: $\forall t \in T$, we assume that $EFT^s(t) < LFT^s(t)$ and that $F_t^s()$ can be expressed as the integral function of a probability density function $f_t^s()$

$$F_t^s(x) = \int_0^x f_t^s(y)dy. \tag{5}$$

The treatment could also be generalized to include immediate and deterministic transitions by resorting to the $\langle Q, U, V\rangle$—partition developed in [13]. However, the assumption of nonpoint-like firing intervals largely simplifies notational conventions. In particular, it guarantees that the probability distribution $F_\tau()$ of every reachable stochastic class is absolutely continuous (provided that this holds in the initial stochastic class Σ^o) and can thus be expressed as the integral function of a probability density function $f_\tau()$. We will thus represent the stochastic class $\langle m, D, F_\tau()\rangle$ as $\langle m, D, f_\tau()\rangle$, and we will call $f_\tau()$ *state-density function*.

3.4 Successors Detection and Calculus of Their Probability

Let $\Sigma = \langle m, D, f_\tau()\rangle$ be a stochastic class, $\langle t_o, t_1, \ldots, t_{N-1}\rangle$ be the vector of transitions enabled by m, and $\tau = \langle \tau_o, \tau_1, \ldots, \tau_{N-1}\rangle$ the vector of their times to fire. t_o is an outcoming event from Σ iff there is a nonempty set of solutions for the *restricted firing domain* D_{t_o} which augments the set D of (3) with the additional constraints $\tau_o \leq \tau_n \forall n \in [1, N-1]$.

According to the semantics of sTPN, the probability μ_{t_o} that t_o is the outcoming event is the probability that the vector of times to fire belongs to D_{t_o} and t_o is selected in the random switch among all transitions which share the same time to fire of t_o. However, since the probability distribution is absolutely continuous, the probability that t_o has the same time to fire of any other transition is equal to 0, and μ_{t_o} is equal to the probability that the vector of times to fire belongs to D_{t_o}

$$\mu_{t_o} = \int_{D_{t_o}} f(x)dx. \tag{6}$$

3.5 Derivation of Successor State-Probability Density Functions

We denote by $\Sigma' = \langle m', D', f'()\rangle$ the stochastic class reached from $\Sigma = \langle m, D, f_Q()\rangle$ through t_o. The marking m' is derived according to (1). The firing domain D' and the probability density function $f'()$ are derived through four steps accounting for: the conditioning introduced by the assumption that t_o is the first transition to fire; the advancement of time observed at the firing of t_o and the elimination of the time to fire of t_o itself; the elimination of times to fire of transitions disabled by the firing of t_o; and the addition of times to fire of transitions newly enabled by the firing of t_o.

Conditioning. The assumption that t_o is the next transition to fire conditions the vector τ, yielding a new random variable τ^a, distributed in the restricted domain D_{t_o} with density function $f_{\tau^a}()$ derived as a conditional probability

$$f_{\tau^a}(x) = \frac{f_Q(x)}{\mu_{t_o}}. \tag{7}$$

Time-advancement. At the firing of t_o, times to fire of all enabled transitions are reduced by τ_o^a and τ_o^a itself is eliminated from the vector of times to fire, yielding a new vector $\tau^b = \langle \tau_1^a - \tau_o^a, \ldots, \tau_{N-1}^a - \tau_o^a\rangle$. The support D^b and the density function $f_{\tau^b}()$ of τ^b are obtained by shifting all the components by τ_o^a and by computing the projection that eliminates τ_o^a

$$\begin{aligned} D^b &= \{\langle x_1, \ldots, x_{N-1}\rangle | \\ &\quad \exists x_o \text{ such that } \langle x_o, x_1 + x_o, \ldots, x_{N-1} + x_o\rangle \in D_{t_o}\}, \\ &f_{\tau^b}(x_1, ..., x_{N-1}) \\ &\quad = \int_{D_o^+(x_1,\ldots,x_{N-1})} f_{\tau^a}(x_o, x_1 + x_o, \ldots, x_{N-1} + x_o)dx_o, \end{aligned} \tag{8}$$

where

$$D_o^+(x_1, \ldots, x_{N-1}) = \{x_o | \langle x_o, x_1 + x_o, \ldots, x_{N-1} + x_o\rangle \in D_{t_o}\}.$$

Disabling. When t_o fires, one or more transitions may be disabled: We denote their number with N_d and we assume that they are encoded in the tail of τ^b. Elimination of disabled transitions yields a new vector of times to fire $\tau^c = \langle \tau_1^b, \ldots, \tau_{N-N_d-1}^b\rangle$, with density function

$$\begin{aligned} D^c &= \{\langle x_1, \ldots, x_{N-N_d-1}\rangle | \\ &\quad \exists \langle x_{N-N_d}, \ldots, x_{N-1}\rangle \text{ such that } \langle x_1, x_2, \ldots, x_{N-1}\rangle \in D^b\}, \\ &f_{\tau^c}(x_1, \ldots, x_{N-N_d-1}) \\ &\quad = \int_{D_d^+(x)} f_{\tau^b}(x_1, \ldots, x_{N-N_d-1}, \ldots, x_{N-1})dx_{N-N_d}, \ldots, dx_{N-1}, \end{aligned} \tag{9}$$

where

$$\begin{aligned} D_d^+(x) = \{&\langle x_{N-N_d}, \ldots, x_{N-1}\rangle | \\ &\langle x_1, \ldots, x_{N-N_d-1}, x_{N-N_d}, \ldots, x_{N-1}\rangle \in D_{\tau^b}\}. \end{aligned}$$

Newly enabling. The vector τ' of times to fire of transitions enabled in $\Sigma' = \langle m', D', f_{\tau'}()\rangle$ is finally obtained by extending τ^c with the times to fire of newly enabled transitions. We denote by N_n the number of newly enabled transitions and we encode them in the tail of the vector of times to fire: $\tau' = \langle \tau_1^c, \ldots, \tau_{N-N_d-1}^c, \tau'_{N-N_d}, \ldots \tau'_{N-N_d-1+N_n}\rangle$. Since the time to fire of each newly enabled transition t is sampled independently according to its static density function $f_t^s()$, if $t_{n(i)}$ denotes the ith newly enabled transition, the density function $f_{\tau'}$ is expressed as

$$\begin{aligned} D' &= D^c \times \prod_{i=0}^{N_n-1} \left[EFT^s(t_{n(i)}), LFT^s(t_{n(i)})\right], \\ &f_{\tau'}(x_1, ..., x_{N-N_d-1}, x_{N-N_d}, ..., x_{N-N_d-1+N_n}) \\ &\quad = f_{\tau^c}(x_1, ..., x_{N-N_d-1}) \cdot \prod_{i=0}^{N_n-1} f_{t_{n(i)}}^s(x_{N-N_d-1+i}). \end{aligned} \tag{10}$$

Authorized licensed use limited to: Universita degli Studi di Firenze. Downloaded on July 2, 2009 at 03:23 from IEEE Xplore. Restrictions apply.

4 Closed-Form Calculus

In this section, we first show that the density function $f_{\tau'}()$ accepts a continuous piecewise representation over a partition of the domain D' in subdomains that are still shaped as DBM zones, and we then develop a closed-form calculus enabling symbolic derivation of the analytic representation of state-density functions for models with expolynomially distributed transitions.

We assume the following notational conventions: If $x = \langle x_1, \ldots, x_{N-1}\rangle \in \mathbb{R}^{N-1}$ and $x_o \in \mathbb{R}$, then $\langle x_o, x\rangle$ denotes the vector $\langle x_o, x_1, \ldots, x_{N-1}\rangle \in \mathbb{R}^N$, and $x + x_o$ denotes the vector $\langle x_1 + x_o, \ldots, x_{N-1} + x_o\rangle \in \mathbb{R}^{N-1}$; besides, according to multi-index notation, for any three real-valued vectors $x, \alpha, \lambda \in \mathbb{R}^N$

$$x^{\alpha} e^{-\lambda x} \stackrel{def}{=} \prod_{n=0}^{N-1} x_n^{\alpha_n} e^{-\lambda_n x_n}. \tag{11}$$

4.1 Piecewise Representation of Density Functions over DBM Subdomains

If the restricted domain D_{t_o} is represented in normal form as $D_{t_o} = \{\tau_i^a - \tau_j^a\} \leq B_{ij}$, then according to (8), $x_o \in D_o^+(x_1, \ldots, x_{N-1})$ iff

$$\max_{i\in[1,N-1]}\{-B_{*o}, -B_{*i} - x_i\} \leq x_o \leq \min_{i\in[1,N-1]}\{B_{o*}, B_{i*} - x_i\}, \tag{12}$$

and $D_o^+(x)$ is thus an interval: $D_o^+(x) = [E_o^+(x), L_o^+(x)]$ with

$$\begin{aligned} L_o^+(x) &= \min_{i\in[1,N-1]}\{B_{o*}, B_{i*} - x_i\}, \quad (a)\\ E_o^+(x) &= \max_{i\in[1,N-1]}\{-B_{*o}, -B_{*i} - x_i\}. \quad (b) \end{aligned} \tag{13}$$

Equation (13) partitions the domain D^b into a finite set of DBM-shaped subdomains within each of which $E_o^+()$ and $L_o^+()$ accept an analytic representation:

Lemma 4.1. *Integration bounds $E_o^+(x)$ and $L_o^+(x)$ are the piecewise continuous functions with analytical representation within each element of a partition of D^b in DBM subdomains R_{ij} with $i, j \in [0, N-1]$. Within each subdomain R_{ij}, $E_o^+()$ and $L_o^+()$ are either constants or monovariate functions of a single component of the vector x*

$$x \in R_{ij} \rightarrow \begin{array}{l} L_o^+(x) = \begin{cases} B_{o*}, & \text{iff } i = 0,\\ B_{i*} - x_i, & \text{iff } i > 0, \end{cases}\\ E_o^+(x) = \begin{cases} -B_{*o}, & \text{iff } j = 0,\\ -B_{*j} - x_j, & \text{iff } j > 0. \end{cases} \end{array} \tag{14}$$

Using Lemma 4.1, we can now characterize the structural properties of state-density functions.

Theorem 4.1. *The state-density function is continuous and accepts a piecewise representation over a set of subdomains in DBM form, provided that the property is satisfied in the initial class, and that static density functions associated with transitions in the model are continuous functions defined by piecewise composition of analytic functions over a finite partition of the static firing interval.*

4.2 Closed-Form Calculus of Density Functions

We develop here a closed-form calculus of state-density functions, under the assumption that static firing intervals of transitions in the sTPN model have expolynomial distributions [39], [16]. This class provides an expressive framework, including common basic distributions (e.g., uniform, exponential, Erlang, and gamma) and enabling fine approximation of other general distributions or approximation of scattered data with limited or unbounded support. The closed-form calculus will show that within each subdomain the state-density function takes the form of a multivariate expolynomial function, and it will make explicit the way in which density functions within state classes grow in complexity as the state accumulates memory through subsequent transitions firings.

By induction, we assume that in each subdomain R of the stochastic class Σ, the density function $f_\tau()$ has the form of a multivariate expolynomial,

$$\begin{aligned} f_\tau^R(x_0, \ldots, x_{N-1}) &= \sum_{h=0}^{H^R-1} c_h^R \cdot \prod_{n=0}^{N^R-1} x_n^{\alpha_{hn}^R} e^{-\lambda_{hn}^R x_n}\\ &= \sum_{h=0}^{H^R-1} c_h^R x^{\alpha_h^R} e^{-\lambda_h^R x}, \end{aligned} \tag{15}$$

and we demonstrate that this form is maintained through the four steps of the derivation of class Σ' reached from Σ through the firing of t_o. For simplicity, we develop the calculus for a single multivariate monomial $c_h^R x^{\alpha_h^R} e^{-\lambda_h^R x}$, without loss of generality as all the steps of the derivation are linear. To reduce notational complexity, indexes h and R will be omitted. According to this, the initial form of the state-density function is expressed as:

$$f_\tau(x_0, x) = c x_o^{\alpha_o} e^{-\lambda_o x_o} x^{\alpha} e^{-\lambda x}. \tag{16}$$

In the *Conditioning* step (cfr. (7)), $f_{\tau^a}()$ is derived by scaling $f_\tau()$ by μ_{t_0}, which maintains the form of (16). In the step of *time-advancement* (cfr. (8))

$$\begin{aligned} f_{\tau^b}(x) &= \int_{E_0^+(x)}^{L_0^+(x)} f_{\tau^a}(x_0, x + x_0)dx_0\\ &= \frac{c \cdot e^{-\lambda x}}{\mu_{t_0}} \int_{E_0^+(x)}^{L_0^+(x)} x_o^{\alpha_o} (x + x_o)^{\alpha} e^{-\Lambda x_o} dx_o, \end{aligned} \tag{17}$$

where $\Lambda = \sum_{n=0}^{N-1} \lambda_n$. By Newton's binomial theorem

$$\begin{aligned} x_o^{\alpha_o}(x + x_o)^{\alpha} &= x_0^{\alpha_0} \prod_{n=1}^{N-1} (x_n + x_0)^{\alpha_n}\\ &= \sum_{f=0}^{F-1} c_f x_0^{\alpha_{0f}} \prod_{n=1}^{N-1} x_n^{\alpha_{nf}}\\ &= \sum_{f=0}^{F-1} c_f x_0^{\alpha_{0f}} x^{\alpha_f}, \end{aligned} \tag{18}$$

where $\alpha_{0f} = \alpha_0 + \sum_{n=1}^{N-1}(\alpha_n - \alpha_{nf})$ and f ranges so as to cover all the possible assignments of the multiindex $\alpha_f = \langle \alpha_{1f}, \ldots, \alpha_{N-1f}\rangle$ which satisfy the constraint

$$0 \leq \alpha_{nf} \leq \alpha_n \quad \forall n \in [1, N-1]. \tag{19}$$

Note that, according to (19), all the monomials in the sum have the same degree, being $\sum_{n=0}^{N-1} \alpha_n = \sum_{n=0}^{N-1} \alpha_{nf}$, and the degree of x_{0f} ranges so as to cover all the possible degrees

Authorized licensed use limited to: Universita degli Studi di Firenze. Downloaded on July 2, 2009 at 03:23 from IEEE Xplore. Restrictions apply.

between the initial degree of x_0 and the maximum degree of the overall polynomial, i.e., $\alpha_{0f} \in \left[\alpha_0, \sum_{n=0}^{N-1} \alpha_n\right]$. Casting (18) into (17), we obtain

$$\begin{aligned} f_{\tau^b}(x) &= \sum_{f=0}^{F-1} c'_f x^{\alpha_f} e^{-\lambda x} \int_{E_0^+(x)}^{L_0^+(x)} x_0^{\alpha_{0f}} e^{-\Lambda x_0} dx_0 \\ &= \sum_{f=0}^{F-1} c'_f x^{\alpha_f} e^{-\lambda x} \big(\Phi_f(L_0^+(x)) - \Phi_f(E_0^+(x))\big), \end{aligned} \tag{20}$$

where $c'_f = \frac{c \cdot c_f}{\mu_o}$ and $\Phi_f(x)$ is the integral function of $x^{\alpha_{0f}} e^{-\Lambda x}$. This takes two different forms depending on whether $\Lambda = 0$ or $\Lambda \neq 0$

$$\begin{aligned} \Phi_f(x) &\stackrel{def}{=} \int x^{\alpha_{0f}} e^{-\Lambda x} dx \\ &= \begin{cases} \dfrac{x^{\alpha_{0f}+1}}{\alpha_{0f}+1}, & \text{if } \Lambda = 0, \\ -e^{-\Lambda x} \displaystyle\sum_{k=0}^{\alpha_{0f}} \frac{\alpha_{0f}!}{k! \Lambda^{1+\alpha_{0f}-k}} \cdot x^k, & \text{if } \Lambda \neq 0. \end{cases} \end{aligned} \tag{21}$$

Remark 4.1. The comparison of (20) and (21) makes explicit a relevant mechanism in the way how the form of state-density functions evolves when the state accumulates memory through subsequent firings. As shown in Lemma 4.1 (cfr. (13)), $L_o^+(x)$ and $E_o^+(x)$ can be constants (B_{o*} and $-B_{*o}$, respectively) or monovariate linear functions of a single component of vector x (in the form of $(B_{i*} - x_i)$ and $(-B_{*j} - x_j)$, respectively). If both $E_o^+(x)$ and $L_o^+(x)$ are constants, then $\Phi_f(L_0^+(x)) - \Phi_f(E_0^+(x))$ is also a constant. Each variable x_n appears with every degree α_{nf} lower than or equal to the initial degree α_n and maintains its initial (possibly null) exponential rate λ_n. Besides, the overall degree $\sum_{n=1}^{N-1} \alpha_{nf}$ of monomials ranges between 0 and $\sum_{n=1}^{N-1} \alpha_n$, which means that the degree of the overall polynomial has been reduced with respect to $f_\tau()$ by the value α_0 of the degree of x_0. Whereas if any of the two bounds, say $E_o^+()$, is a linear function of x_i, then $\Phi_f(E_0^+(x))$ becomes a function of x_i that inherits the exponential rate and the polynomial degree of $\Phi_f()$ of (21): If $\Lambda = 0$ (i.e., if the initial form of f_τ does not contain any exponential factor), then the maximal polynomial degree of $f_{\tau^c}()$ increases by 1 with respect to $f_\tau()$ and x_i will appear with a maximum degree of $1 + \sum_{n=0}^{N-1} \alpha_n$; if $\Lambda \neq 0$, then the maximal polynomial degree of f_{τ^c} is equal to that of $f_\tau()$, but the exponential rate of x_i is increased by Λ (i.e., by the sum of all exponential rates) and x_i will appear with all the degrees in the interval $[0, \sum_{n=0}^{N-1} \alpha_n]$, always with exponential rate $(\Lambda - \lambda_i)$.

In the step of *disabling* (cfr. (9)), $f_{\tau^b}()$ is repeatedly integrated to marginalize the distribution with respect to the time to fire of each disabled transition. Also in this case, if both integration bounds $E_o^+()$ and $L_o^+()$ are constants, the polynomial degree and the rate of the marginalized component are lost. Whereas if any of the two bounds is a linear function of a component, say x_i, then rate and polynomial degree are transferred to x_i. Also in this case, the maximal polynomial degree grows by 1 if no exponential factors are present. As a major difference with respect to the step of time-advancement, disabling does not involve a time shift. This avoids the Newton binomial expansion of each monomial that occurred in the step of time-advancement.

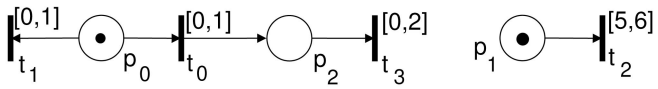


Fig. 1. A simple sTPN model. Static density functions of all transitions are supposed to be uniform.

In the step of *newly enabling* (cfr. (10)), the state-density function $f'_\tau()$ of the class Σ' is derived by multiplying $f_{\tau^c}()$ by the static density functions of newly enabled transitions. Note that in this step, preservation of the multivariate expolynomial form of the density function depends on the assumption that static density functions of transitions in the model are (univariate) expolynomial functions.

4.3 Example of Calculus

To help the understanding, we exemplify the calculus with reference to the net in Fig. 1.

Initial stochastic class. Three transitions are initially enabled: t_o, t_1, and t_2. We assume that their times to fire are thus independent and distributed according their respective (uniform) static density functions, as if they were newly enabled. The initial stochastic class Σ^o is thus

$$\begin{aligned} &m_o = p_o p_1, \\ &\tau = \langle \tau_o, \tau_1, \tau_2 \rangle, \\ &f_\tau(x_0, x_1, x_2) = \tfrac{1}{\|D\|} = 1, \end{aligned} \qquad D = \begin{cases} 0 \le \tau_0 \le 1 \\ 0 \le \tau_1 \le 1 \\ 5 \le \tau_2 \le 6. \end{cases} \tag{22}$$

Successor detection and probability. t_o is a possible outcoming event as the restricted domain D_{t_o} imposing τ_o to be minimum is not empty. Its probability is $\frac{1}{2}$

$$\begin{aligned} D_{t_0} &= \begin{cases} D \\ \tau_0 - \tau_1 \le 0 \\ \tau_0 - \tau_2 \le 0, \end{cases} \\ \mu_{t_0} &= \int_{D_{t_0}} f_\tau(x_0, x_1, x_2) dx_0 dx_1 dx_2 = \frac{1}{2}. \end{aligned} \tag{23}$$

Conditioning. The assumption that t_0 fires first conditions the vector of times to fire, yielding a new random variable $\tau^a = \tau | (\tau_0 \le \tau_1) \wedge (\tau_0 \le \tau_2)$, which ranges over the restricted domain D^a with density function $f_{\tau^a}()$

$$D^a = \begin{cases} 0 \le \tau_0^a \le 1 \\ 0 \le \tau_1^a \le 1 \\ 5 \le \tau_2^a \le 6 \\ \tau_0^a - \tau_1^a \le 0 \\ \tau_0^a - \tau_2^a \le 0 \end{cases} \quad f_{\tau^a}(x_0, x_1, x_2) = \frac{f_\tau(x_0, x_1, x_2)}{\mu_{t_0}} = 2. \tag{24}$$

Time-advancement. At the firing of t_0, according to (8), the vector of times to fire τ^b is derived by reducing by τ_0^a all the times to fire in τ^a and by computing the projection that eliminates τ_0^a. To illustrate the different effects of the reduction and the projection, we split the derivation in

Authorized licensed use limited to: Universita degli Studi di Firenze. Downloaded on July 2, 2009 at 03:23 from IEEE Xplore. Restrictions apply.

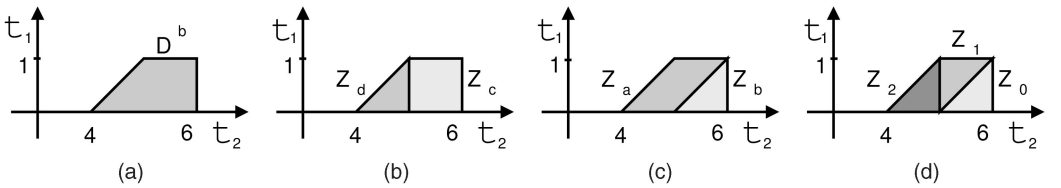


Fig. 2. (a) Domain D^b and its partition induced by (b) $E_o^+()$, (c) $L_o^+()$, and (d) by their combination.

two steps: we first derive a technical intermediate variable $\tau^{b'} =< \tau_0^a, \tau_1^a - \tau_0^a, \tau_2^a - \tau_0^a >$, and we then compute τ^b as the projection that eliminates the first component of $\tau^{b'}$. The intermediate variable $\tau^{b'}$ ranges over $D^{b'}$ with density function $f_{\tau^{b'}}()$.

$$D^{b'} = \begin{cases} 0 \le \tau_0^{b'} \le 1 \\ 0 \le \tau_1^{b'} + \tau_0^{b'} \le 1 \\ 5 \le \tau_2^{b'} + \tau_0^{b'} \le 6 \\ 0 \le \tau_1^{b'} \le 1 \\ 4 \le \tau_2^{b'} - \tau_1^{b'} \le 6 \\ 4 \le \tau_2^{b'} \le 6, \end{cases} \tag{25}$$

$$f_{\tau^{b'}}(x_0, x_1, x_2) = f_{\tau^a}(x_0, x_1 + x_0, x_2 + x_0) = 2.$$

Remark 4.2. The fact that $f_{\tau^a}()$ is a constant hides part of the effect of the shift. If we had a more complex function, say $f_{\tau^a}(x_0, x_1, x_2) = c \cdot x_0^{n_0} \cdot x_1^{n_2} \cdot x_2^{n_2} \cdot e^{-\lambda_0 x_0 - \lambda_1 x_1 - \lambda_2 x_2}$, then the shift would yield the form

$$f_{\tau^{b'}}(x_0, x_1, x_2) = c x_0^{n_0} (x_1 + x_0)^{n_2} (x_2 + x_0)^{n_2} \cdot e^{-(\lambda_0 + \lambda_1 + \lambda_2) x_0 - \lambda_1 x_1 - \lambda_2 x_2}.$$

In this case, the multivariate monomial would explode into a polynomial summation where x_0 appears with every degree between n_0 and $n_0 + n_1 + n_2$ (Newton binomial theorem). In particular, we would find a monomial term of the form $\cdots + c \cdot x_0^{n_0+n_1+n_2} \cdot e^{-(\lambda_0+\lambda_1+\lambda_2)x_0} + \cdots$, where x_0 "inherits" the rate and the monomial degree of x_1 and x_2.

The vector τ^b is derived from $\tau^{b'}$ through the projection that eliminates $\tau_o^{b'}$. According to this, τ^b ranges over the projection of $D^{b'}$ (see Fig. 2a)

$$D^b = \begin{cases} 0 \le \tau_1^b \le 1 \\ 4 \le \tau_2^b - \tau_1^b \le 6 \\ 4 \le \tau_2^b \le 6. \end{cases} \tag{26}$$

The density function of τ^b is obtained by marginalizing $f_{\tau^{b'}}()$ with respect to $\tau_0^{b'}$

$$\begin{aligned} f_{\tau^b}(x_1, x_2) &= \int_{E_0^+(x_1,x_2)}^{L_0^+(x_1,x_2)} f_{\tau^a}(x_0, x_1 + x_0, x_2 + x_0) dx_0 \\ &= 2(L_0^+(x_1, x_2) - E_0^+(x_1, x_2)). \end{aligned} \tag{27}$$

Remark 4.3. Also in this case, the fact that $f_{\tau^{b'}}()$ is a constant hides part of the effects. If we have a more complex function $f_{\tau^{b'}}(x_0, x_1 + x_0, x_2 + x_0) = \cdots + g(x_1, x_2) \cdot x_0^{n_0}$, then $f_{\tau^b}()$ takes the form

$$\begin{aligned} f_{\tau^b}(x_1, x_2) &= \cdots + g(x_1, x_2) \int_{E_0^+(x_1,x_2)}^{L_0^+(x_1,x_2)} x_0^{n_0} dx_0 \\ &= \cdots - g(x_1, x_2) \left[\frac{x_0^{n_0+1}}{n_0 + 1} \right]_{E_0^+(x_1,x_2)}^{L_0^+(x_1,x_2)}. \end{aligned} \tag{28}$$

Besides, if $f_{\tau^{b'}}(x_0, x_1, x_2)$ also includes an exponential factor for x_o such as in $f_{\tau^{b'}}(x_0, x_1, x_2) = \cdots + g(x_1, x_2) \cdot x_0^{n_0} \cdot e^{-\lambda_0 x_0}$, then $f_{\tau^b}()$ takes the form

$$\begin{aligned} f_{\tau^b}(x_1, x_2) &= \cdots + g(x_1, x_2) \int_{E_0^+(x_1,x_2)}^{L_0^+(x_1,x_2)} x_0^{n_0} e^{-\lambda_0 x_0} dx_0 \\ &= \cdots - g(x_1, x_2) \left[e^{-\lambda_0 x_0} \sum_{k=0}^{n_0} c_k x_0^k \right]_{E_0^+(x_1,x_2)}^{L_0^+(x_1,x_2)}. \end{aligned} \tag{29}$$

Note that, while in (28), the monomial degree of x_0 is incremented by 1; in (29), rate and maximum monomial degree of x_o are held unchanged.

Coming back to (27), $E_0^+(x_1, x_2)$ and $L_0^+(x_1, x_2)$ denote the maximum and the minimum among all the functions that lower bound and upper bound $\tau_0^{b'}$, respectively. By replacing $\tau_1^{b'} = x_1$ and $\tau_2^{b'} = x_2$ in the expression of $D^{b'}$ (see (25)), we obtain that the range of variability for $\tau_o^{b'}$ is

$$Max\{0, -x_1, 5 - x_2\} \le \tau_o^{b'} \le Min\{1, 1 - x_1, 6 - x_2\}, \tag{30}$$

which means that $E_0^+(x_1, x_2) = Max\{0, -x_1, 5 - x_2\}$ and $L_0^+(x_1, x_2) = Min\{1, 1 - x_1, 6 - x_2\}$. Equation (30) partitions domain D^b in two subdomains where $E_0^+()$ takes different analytical forms: $Max\{0, -x_1, 5 - x_2\}$ is equal to $5 - x_2$ iff $\langle x_1, x_2 \rangle \in D^b \wedge (5 - x_2 \ge 0) \wedge (5 - x_2 \ge -x_1)$, which defines DBM zone Z_d (see Fig. 2b); in a similar manner, $Max\{0, -x_1, 5 - x_2\}$ is equal to 0 iff $\langle x_1, x_2 \rangle \in D^b \wedge 0 \ge -x_1 \wedge 0 \ge 5 - x_2)$, which defines a DBM zone Z_c; in no part of D^b, $Max\{0, -x_1, 5 - x_2\}$ is equal to $-x_1$. Besides, (30) partitions D^b in two further DBM subdomains Z_a and Z_b where $L_0^+()$ takes different analytical forms (see Fig. 2c). The combination of partitions Z_a, Z_b and Z_c, Z_d yields a final refined partition of D^b in three DBM subdomains Z_1, Z_2 and Z_3 within each of which both $E_0^+()$ and $L_0^+()$ accept analytic representation (see Fig. 2d). In conclusion, $f_{\tau^b}(x_1, x_2)$ can be expressed in piecewise form as

$$\begin{aligned} f_{\tau^b}(x_1, x_2) &= 2 \cdot (L_0^+(x_1, x_2) - E_0^+(x_1, x_2)) \\ &= \begin{cases} 2 \cdot (1 - x_1) & \text{if } \langle x_1, x_2 \rangle \in Z_0 \\ 2 \cdot (6 - x_2) & \text{if } \langle x_1, x_2 \rangle \in Z_1 \\ 2 \cdot (x_2 - x_1 - 4) & \text{if } \langle x_1, x_2 \rangle \in Z_2. \end{cases} \end{aligned} \tag{31}$$

Note that, since each subdomain of D^b includes at least one diagonal cut (i.e., an upper or lower bound expressed as a

Authorized licensed use limited to: Universita degli Studi di Firenze. Downloaded on July 2, 2009 at 03:23 from IEEE Xplore. Restrictions apply.

linear function of x_1 or x_2) and since the form of $f_\tau()$ in (22) does not include any exponential term, the maximum degree of $f_{\tau^b}()$ turns out to be increased by 1 with respect to $f_\tau()$ all over D^b.

Remark 4.4. It is worth stressing the fact that the form of equations defining subdomains depends on the form of the domain D^b, and not on the density function $f_{\tau^b}()$. According to this, the same partition in subdomains would be maintained for any different initial form of $f_\tau()$.

In the model of Fig. 1, when t_0 fires, t_1 is disabled. According to this, the step of *disabling* yields a new monovariate vector of times to fire τ_c, which ranges over the domain $D^c = \{4 \le \tau_2^c \le 6$. (obtained by projecting D^b on τ_2^b), with density function

$$f_{\tau^c}(x_2) = \int_{E_1^+(x_2)}^{L_1^+(x_2)} f_{\tau^b}(x_1, x_2)dx_1.$$

Derivation of the integral form is much similar to the previous step, as in this case, there is not a shift, while derivation of integration bounds is exactly the same problem as in the previous step and yields a similar problem of partition.

In the final step of *newly enabling*, according to the model of Fig. 1, at the firing of t_0, transition t_3 is newly enabled. This yields the final variable $\tau' =< \tau_2^c, \tau_3' >$, where the new component τ_3 is independent and distributed according to its static density function. τ' will thus range over the domain D' with density function expressed in the form of product

$$D' = \begin{cases} 4 \le \tau_2' \le 6 \\ 0 \le \tau_3' \le 2, \end{cases} \quad f_{\tau'}(x_2, x_3) = f_{\tau'}(x_2) \cdot f_{t_3}^s(x_2). \qquad (32)$$

5 Approximate State-Space Representation

Application of the analysis based on stochastic classes is limited by two factors of different nature. In the stochastic expansion, domain partitioning and polynomial degree grow with the degree of concurrency and with the length of behaviors in which transitions overlap their activities [33]. This exacerbates the state explosion problem of nondeterministic analysis, affecting not only the number of stochastic classes but also the cost of their derivation.

Besides, the analysis may even be infeasible because the stochastic expansion does not terminate. A theory that guarantees termination is developed in [13] by leveraging on the concept of *regeneration class*: *1)* a state class is a regeneration class when all its timers are either newly enabled, or they are exponentially distributed, or they are deterministic or deterministically dependent on a timer satisfying any of the previous conditions and *2)* if every cycle in the graph of state classes visits at least one regeneration class, then stochastic expansion is bounded. The condition is in some sense intrinsic to the problem more than to the analysis technique: if the graph of state classes includes a cycle that never visits a regeneration class, then the model accepts a firing sequence that never reaches a state that satisfies the Markov property. Yet, a large variety of models with applicative meaning do not satisfy the condition.

We attack both the problems through two complementary techniques of approximate representation of the state space: relaxation of the equivalence between state-density functions reduces the number of stochastic classes expanding the same state classes; replacement of state-density functions with an approximant accepting a global and structurally invariant analytic representation over the entire domain guarantees a constant level of complexity in the derivation of successor classes.

5.1 Product Decomposition of State-Density Functions

The closed-form calculus of Section 4 shows that not all the components of state-density functions contribute to the growth of polynomial degree and domain partitioning. According to (10), for each newly enabled transition t_i, the time to fire τ_i is distributed independently according to the static density function $f_{t_i}^s()$. Besides, the time to fire of an exponentially distributed transition t_e always maintains an exponential distribution independent of any other time to fire.

Lemma 5.1. *if* $[EFT^s(t_e), LFT^s(t_e)] = [0, \infty]$ *and* $f_{t_e}^s(x_e) = \lambda_e e^{-\lambda_e x_e}$, *then in every stochastic class* Σ *where* t_e *is enabled, the firing domain and the state-density function can be expressed as*

$$D_{\langle \tau_e, \tau \rangle} = \begin{cases} D_\tau \\ \tau_e - \tau_* \ge 0 \end{cases} \quad f_{\langle \tau_e, \tau \rangle}(x_e, x) = \lambda_e e^{-\lambda_e x_e} f_\tau(x), \quad (33)$$

where $\langle \tau_e, \tau \rangle$ *denotes the vector of times to fire of enabled transitions, and* D_τ *is the projection of* $D_{\langle \tau_e, \tau \rangle}$ *on the subspace of* τ.

The state-density function of any class can thus be expressed as the product of three factors separating exponential, newly enabled, and other persistent variables. We thus decompose the vector of times to fire in three subvectors $\tau = \langle \alpha, \beta, \gamma \rangle$ which encode the times to fire of persistent generally distributed transitions, of newly enabled generally distributed transitions, and of exponential transitions, respectively. We denote by G, N, and E the sizes of α, β, and γ, and we assume that $\nu(n)$ returns the index of the nth newly enabled transition, and $\epsilon(e)$ returns the index of the eth exponentially distributed transition. Under this notation:

Theorem 5.1. *In every reachable stochastic state class* $\Sigma = \langle m, D, f_\tau() \rangle$, *the firing domain* D *and the state-density function* $f_\tau()$ *can be expressed as*

$$f_{\langle \alpha, \beta, \gamma \rangle}(x, y, z) = f_\alpha(x) \cdot \prod_{n=0}^{N-1} f_{t_{\nu(n)}}^s(y_n) \cdot \prod_{e=0}^{E-1} \lambda_{\epsilon(e)} e^{-\lambda_{\epsilon(e)} z_{\epsilon(e)}},$$

$$D_{\langle \alpha, \beta, \gamma \rangle} = D_\alpha \times \left(\prod_{n=0}^{N-1} [EFT^s(t_{\nu(n)}), LFT^s(t_{\nu(n)})] \right) \times [0, \infty]^E. \qquad (34)$$

According to (34), subvectors β and γ are uniquely determined by the underlying state class and do not contribute to the complexity of either the domain partitioning or the polynomial complexity of density functions. According to this, relaxation of the test of equivalence and

Authorized licensed use limited to: Universita degli Studi di Firenze. Downloaded on July 2, 2009 at 03:23 from IEEE Xplore. Restrictions apply.

approximation in the representation of state-density functions are conveniently applied only to the subvector α.

5.2 Relaxed Equivalence between State-Density Functions

The relation of equivalence between state-density functions can be relaxed by assuming that two stochastic classes are equivalent as soon as they have the same underlying state class (i.e., the same marking and domain), and the distance d between their state-density functions measured according to some suitable metrics d is below a given threshold of tolerance δ

$$\langle m, D, f() \rangle =_\delta \langle m, D, g() \rangle \quad \text{iff} \quad \|f() - g()\|_d \leq \delta. \tag{35}$$

In particular, to exploit the unit-measure property of density functions, we assume

$$\|f() - g()\|_d \stackrel{def}{=} \frac{1}{D} \int_D |f(x) - g(x)| dx. \tag{36}$$

Note that since $f()$ and $g()$ are positive and have unit measure over D, then $\|f() - g()\|_d \in [0, 2]$.

5.3 Approximation with Bernstein Polynomials

Relaxation of the test of equivalence guarantees that the stochastic expansion does not diverge in the enumeration of an infinite sequence of state-density functions that converge to some limit but always have different domain partition and analytic form. However, this relaxation does not cope with the problem of stochastic classes with a growing complexity in domain partitioning and/or in the polynomial degree of state-density functions. A complementary technique is thus developed, which introduces a mechanism of approximation in the symbolic representation of state-density functions.

To this end, we further restrict the class of static density functions of model transitions by assuming that transitions with unbounded support are all distributed over $[0, \infty]$ with (negative) exponential distribution. This restriction guarantees that D_α is bounded and thus compact. Since Theorem 4.1 guarantees that $f_\alpha()$ is a continuous function, by Weierstrass theorem $f_\alpha()$ is bounded and can be arbitrarily well approximated by polynomials. Note that, as a fortunate fact, polynomials can be managed in the closed-form calculus of Section 4, which can thus be used to derive the successors of a class with an approximated density function.

As a specific kernel of approximant functions, we resort to multivariate Bernstein Polynomials [26], [34]. In their basic formulation, these approximate a function of G variables defined over a compact hyperrectangular domain, by weighting a kernel of multivariate monomials according to the samples of the approximated function taken over a regular grid. In particular, for a function $f_\alpha()$ of G variables ranging over the hypercube $[0, 1]^G$, a grid that takes K_g samples for each variable x_g yields the following approximant $\tilde{f}_\alpha()$:

$$\begin{aligned} \tilde{f}_\alpha(x_0, \ldots, x_{G-1}) = & \sum_{\substack{0 \leq k_g \leq K_g - 1 \\ g \in [0, G-1]}} f_\alpha\left(\frac{k_o}{K_o}, \ldots, \frac{k_{G-1}}{K_{G-1}}\right) \\ & \cdot \prod_{g=0}^{G-1} \binom{K_g}{k_g} x_g^{k_g} (1 - x_g)^{K - k_g}, \end{aligned} \tag{37}$$

where $\sum_{\substack{0 \leq k_g \leq K_g - 1 \\ g \in [0, G-1]}}$ denotes the multiple summation

$$\sum_{k_0=0}^{K_0-1} \sum_{k_1=0}^{K_1-1} \cdots \sum_{k_{G-1}=0}^{K_{G-1}-1}.$$

Approximation based on Bernstein polynomials exhibits a number of properties that nicely fit the needs of our application.

Globality. The approximant is global, in the sense that it is analytic over the entire domain of the approximated function. This avoids piecewise composition and partition in subdomains, as would be required by other approximant functions.

Positivity. Since all polynomials in the kernel are positive over the entire domain and all the samples of the state-density function are positive, the approximant is still positive over the entire domain.

Simplicity of derivation. The form of Bernstein approximant is determined in straightforward manner by the samples of the approximated function, without requiring any optimization process which could not be afforded in the core steps of state-space enumeration.

Convergence. If the approximated function is continuous, and this is the case of our state-density function according to Theorem 4.1, the approximant converges uniformly to the approximated function when $\sum_{g=0}^{G-1} \frac{1}{K_g} \to 0$. More specifically, the approximation error can be bounded through a Lipschitz inequality

$$\begin{aligned} & \|f_\alpha(x) - f_\alpha(y)\|_2 \leq L \cdot \|x - y\|_2 \\ & \quad \to \|\tilde{f}_\alpha(x) - f_\alpha(x)\|_2 \leq \frac{L}{2} \left(\sum_{g=0}^{G-1} \frac{1}{K_g} \right)^{\frac{1}{2}}. \end{aligned} \tag{38}$$

Conversely, other requirements are not guaranteed in a straightforward manner but can be accommodated through a suitable adaptation.

Unit measure. Bernstein approximation does not guarantee that the approximant preserves the unit-measure property of the approximated state-density function, though we can expect that a good approximant of a unit-measure function is still close to have unit measure. To manage the problem, after the approximant has been computed, we normalize it with respect to its own integral over the domain.

Equivalence. Since we introduce an approximation, the test of equivalence between stochastic classes must necessarily be relaxed according to (36). For the sake of efficiency, the metric $\|\cdot\|_d$ itself is approximated in discrete form by evaluating it in correspondence with samples. Since in the application, the number of samples K_g may be relatively small, the approximation takes into account border effects on the grid:

Authorized licensed use limited to: Universita degli Studi di Firenze. Downloaded on July 2, 2009 at 03:23 from IEEE Xplore. Restrictions apply.

$$\|f()\|_d \simeq \frac{\sum_{\substack{0\le k_g\le K_g-1\\ g\in[0,G-1]}} A(k_o,\ldots,k_{G-1})\cdot\left|f\left(\frac{k_o}{K_o},\ldots,\frac{k_{G-1}}{K_{G-1}}\right)\right|}{\sum_{\substack{0\le k_g\le K_g-1\\ g\in[0,G-1]}} A(k_o,\ldots,k_{G-1})}, \tag{39}$$

where $A(k_o,\ldots,k_{G-1})=\frac{1}{2^{border(k_o,\ldots,k_{G-1})}}$, with $border(k_o,\ldots,k_{G-1})$ denoting the number of elements of $\langle k_o,\ldots,k_{G-1}\rangle$ that are equal to 0 or K_g.

DBM-domain. The formulation of Bernstein polynomials can be easily extended to the case of a function $f()$ defined over a generic hyperrectangle $D=\prod_{g=0}^{G-1}[a_g,b_g]$, which comprises an affine transformation of the hypercube of (37). In particular, if b_{ij} denote the coefficients of the normal form of D_α and $a_g=-b_{*g}$ and $b_g=b_{g*}$, then the Bernstein approximant defined over the minimum embedding hyperrectangle of D_α is

$$\begin{aligned}\tilde{f}_\alpha(x_0,\ldots,x_{G-1}) &= \sum_{\substack{0\le k_g\le K_g-1\\ g\in[0,G-1]}} f_\alpha\Bigg(a_o+\frac{k_o}{K_o}(b_o-a_o),\ldots,\\ &\qquad a_{G-1}+\frac{k_{G-1}}{K_{G-1}}(b_{G-1}-a_{G-1})\Bigg)\\ &\cdot\prod_{g=0}^{G-1}\binom{K_g}{k_g}\frac{1}{(b_g-a_g)^{K_g}}(x_g-a_g)^{k_g}(b_g-x_g)^{K_g-k_g}.\end{aligned} \tag{40}$$

In this expression, the approximant function $\tilde{f}_\alpha()$ depends on samples that belong to the minimum embedding hyperrectangle of D_α, denoted by R_α, but not to D_α itself. These samples can be given arbitrary (nonnegative) values, provided that the condition of Lipschitz continuity in the left-hand side of (38) is preserved. Since D_α is convex, for any $x\in R_\alpha\setminus D_\alpha$, there exists uniquely a point $\bar{x}\in D_\alpha$ that minimizes the distance from x; if we assign to $f_\alpha(x)$ the value $f_\alpha(\bar{x})$, we obtain an extension of $f_\alpha()$ over the entire hyperrectangle R_α that satisfies Lipschitz continuity. Unfortunately, the selection of $\bar{x}$ amounts to a problem of quadratic optimization over a convex linear polyhedron, which cannot be afforded for the reasons of efficiency. We thus assume a simplified assignment that extends $f_\alpha()$, without attaining the minimum distance. The assignment is obtained through Algorithm 8.1 defined in the Appendix: using the normal form of D_α, the algorithm repeatedly corrects individual components of the vector x so that after g corrections, all the constraints involving any of the first g components are satisfied; as salient traits, the algorithm maps close points to close approximants so as to maintain Lipschitz continuity, and it can be embedded in the code that tests the condition $x\in D_\alpha$ without increasing its computational complexity.

6 Computational Experience

We report experimental results that compare limits of applicability and accuracy of results in the enumeration of the stochastic class graph based on: the exact implementation of the closed form calculus (called *exact-analysis*); the variant that uses relaxed equivalence (*relaxed-analysis*); the variant that jointly uses relaxed equivalence and Bernstein approximation (*Bernstein-analysis*). Tests are repeated with values 0.5, 0.05, and 0.005 for the approximation threshold δ of (35) (called threshold), and with values 2, 3, 4, 5, 6 for the maximum degree K_g of (37) applied to each variable in Bernstein approximation (called degree) which we set equal for all the variables of the approximated state-density function. Experimentation refers to a set of models that incrementally stress first the limits of time and space and then the condition of theoretical unboundedness. The complexity of analysis was evaluated in terms of various metrics, and notably the number of enumerated stochastic classes and a qualitative measure of time spent in the enumeration. The accuracy maintained in the approximation was evaluated by comparing steady-state probabilities in the Discrete-Time embedded Markov Chain comprised by the stochastic class graph [13] derived through different analysis techniques. In particular, we compared the vector of probabilities of reachable markings, which by construction are preserved with relative robustness across different approximations. In the assessment, we compare the overall fit, together with average and maximum difference in the probability of the same marking in any two different analysis techniques. We also compare against simulation results obtained through a stochastic simulator implemented in the Oris tool. In the methodology of experimentation, this serves both to assess the exact-analysis method and provide a reference of comparison for results obtained on models that are not amenable to exact-analysis.

6.1 Three-Tasks Model

Fig. 3 reports the sTPN of a model called *Three-Tasks*. This represents three concurrent cyclic tasks (formed by transitions t1x, t2x, and t3x, respectively), sharing three mutually exclusive resources (places R12, R23, and R31) and synchronizing at the end of each cycle (transition restart). Each resource acquisition and each operation requires a time uniformly distributed in [0,100] and [200,400], respectively. Transition restart is exponentially distributed over $[0,\infty]$ with rate 0.003.

Nondeterministic analysis [37] yields 228 state classes in 0.1 seconds. Stochastic expansion is by far out of the range of feasibility for the implementation that delegates to Mathematica the symbolic calculus [32], but can be successfully completed through exact-analysis, relaxed-analysis, and Bernstein-analysis. *Exact-analysis* yields 1,903 stochastic classes, with a maximum internal complexity of 25 subdomains in the same class and a maximum polynomial degree of 13, in 58 seconds. *Relaxed-analysis* yields 214 stochastic classes with threshold 0.005 and 210 classes with thresholds 0.05 and 0.5, in times ranging within 11 and 12 seconds. Note that the number of stochastic classes is basically insensitive to the threshold; also note that stochastic classes are less than nondeterministic classes as these also include classes that are reached through null probability transitions, which are suppressed in stochastic expansion (the minimum accepted transition probability was set equal to 10^{-40}). *Bernstein-analysis* yields 210 classes for all values of the threshold and the degree. In general, by construction, the number of classes is affected by the

Authorized licensed use limited to: Universita degli Studi di Firenze. Downloaded on July 2, 2009 at 03:23 from IEEE Xplore. Restrictions apply.

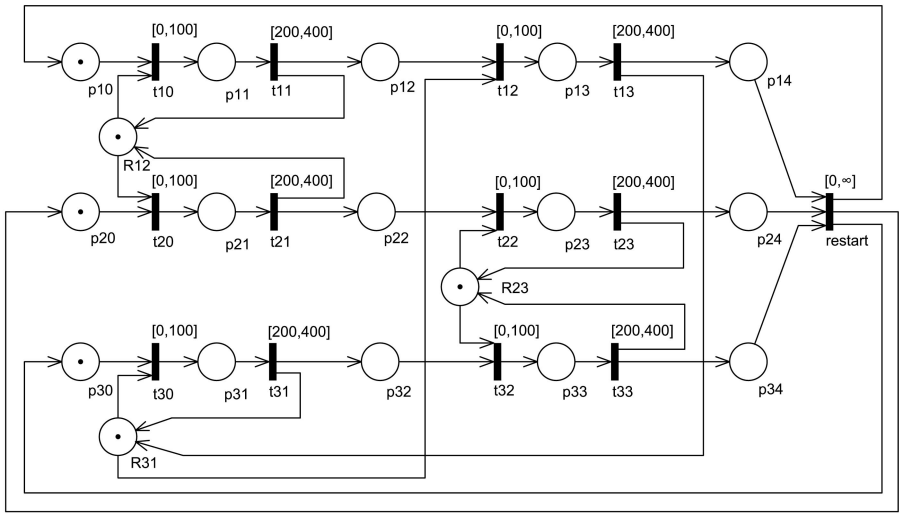


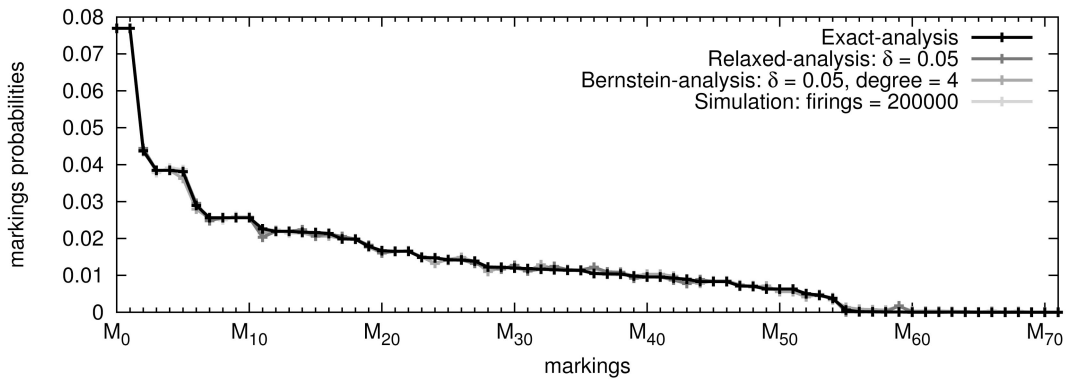
Fig. 3. Three-Tasks model: An sTPN model representing three recurrent tasks. Times to fire are uniformly distributed within their static intervals, with the sole exception of transition restart, which is exponentially distributed over $[0, \infty]$ with rate 0.003.

threshold much more than by the degree; and in this case, the model is quite insensitive to the threshold as observed in relaxed-analysis. Instead, enumeration time is equal to 1, 2, 4, 8, and 14 seconds for degrees 2, 3, 4, 5, and 6, respectively, and it is not affected at all by the threshold. In general, enumeration time primarily depends on the number of classes, which, in turn, may depend on the threshold. *Simulation* takes about 10 hours to complete 200,000 firings.

Fig. 4 compares steady-state probabilities of reachable markings obtained with the different analysis techniques and simulation. Average errors indicate that *simulation* performs better than *relaxed-analysis* and this performs better than *Bernstein-analysis*. However, consider that the average error is in any case lower than 0.0028 and Bernstein-analysis is completed in 4 seconds, relaxed-analysis in 12 seconds, while simulation results refer to an experiment of more than 10 hours.

Systematic experimentation was carried out to observe the variation of each technique with respect to its specific parameters. *Simulation* depends, of course, on the number of firings, yielding an average error equal to 0.00120, 0.00067, 0.00018, and 0.00017 for a number of firings equal to 1,000, 10,000, 100,000, and 200,000, respectively, with experimentation time varying in a basically linear manner. *Relaxed-analysis* is insensitive to the variation of the threshold, which could be expected in this model where the threshold does not affect the number of stochastic classes. *Bernstein-analysis* is of course equally insensitive to the threshold, but varies with the degree, yielding an average error of 0.00062, 0.00053, 0.00041, 0.00036, and 0.00034 with degrees equal to 2, 3, 4, 5, and 6, respectively.

The Three-Tasks model can be modified by replacing all uniform distributions with expolynomial forms: each resource acquisition and each task operation require a time distributed within $[0, 100]$ and $[200, 400]$, respectively, according to density functions $f_{wait}(x) = k_{wait} x e^{-0.003x}$ and $f_{op}(x) = k_{op}(x + x e^{-0.003x})$, respectively (where k_{wait} and k_{op} are normalization factors and mean time is 64.96756 and 308.32637, respectively).



$\langle average, max \rangle$	Relaxed-analysis: $\delta = 0.05$	Bernstein-analysis: $\delta = 0.05$, deg=4	Simulation: 200,000 firings
Exact	$\langle 0.00028, 0.00231 \rangle$	$\langle 0.00041, 0.00183 \rangle$	$\langle 0.00017, 0.00061 \rangle$

Fig. 4. Three-Tasks model: (Top) Measures of markings probabilities with exact-analysis, relaxed-analysis (with threshold 0.05), Bernstein-analysis (with threshold 0.05 and degree 4), and simulation (with 200,000 firings); (bottom) average and maximum deviation in measures of different techniques.

Authorized licensed use limited to: Universita degli Studi di Firenze. Downloaded on July 2, 2009 at 03:23 from IEEE Xplore. Restrictions apply.

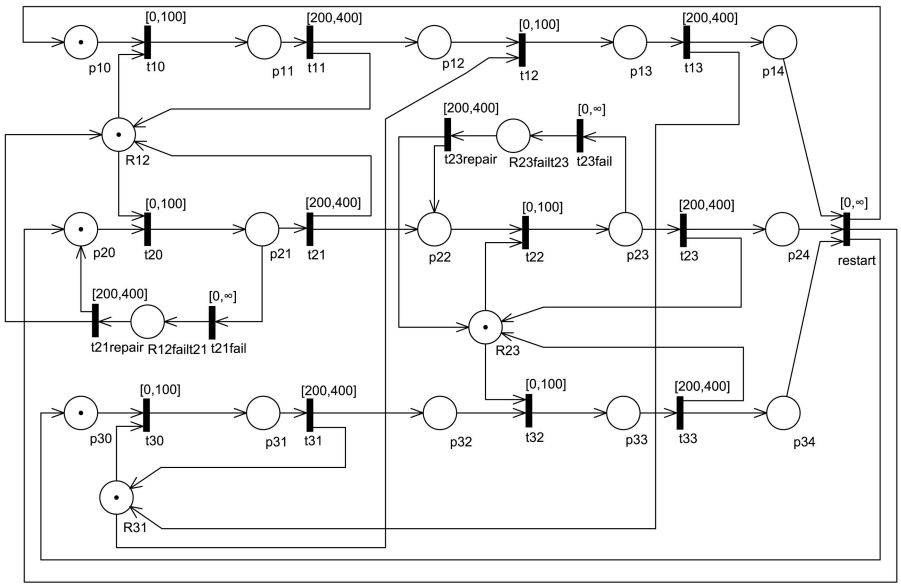


Fig. 5. (Fail&Repeat) A variant of the Three-Tasks model including failures and repair actions for resources used by the mid task. Failed actions are repeated.

The variation impacts on the complexity of exact and relaxed analysis, whereas it does not basically affect complexity and accuracy of Bernstein-approximated analysis, which thus obtains a relevant gain in complexity. *Exact-analysis* enumerates 1,914 stochastic classes in a little more than 2 hours and a half. *Relaxed-analysis* is nearly independent of the threshold and yields a number of stochastic classes between 214 and 210 in a little more than 20 minutes. The average error with respect to exact analysis is 0.00041, and the maximum is 0.00349, both insensitive to the threshold. These values compare against average 0.00028 and maximum 0.00231 observed in the Three-Tasks model. *Bernstein-analysis* yields a number of stochastic classes ranging within 210 and 205; enumeration time is not affected by the threshold and varies from 6 to 59 seconds when the degree varies from 2 to 6; this marks a minor increase with respect to the time ranging within 1 and 14 seconds observed in the Three-Tasks model. With degree 4 and threshold 0.005, the average error with respect to the exact analysis is 0.00058 and the maximum is 0.00244. These compare against average 0.00041 and maximum 0.00183 observed in the Three-Tasks model.

6.2 Fail&Repeat Model

Fig. 5 describes a different variant of the Three-Tasks model, which we call *Fail&Repeat* model, where resources R12 and R23 may fail. Failures occur with exponential distribution with rate 0.003 during the period in which they are used by the mid task modeled by transitions t20, t21, t22, and t23. Failed resources are repaired in time uniformly distributed in the interval [200,400], and the failed action is repeated after repair.

Nondeterministic analysis yields 839 state classes in nearly half a second. The stochastic class graph is guaranteed to be finite as the model is regenerative [13]: Any cyclic behavior includes a firing of restart; since failed actions are repeated, at the firing of restart, no transition is persistent. However, both *exact-analysis* and *relaxed-analysis* do not complete the enumeration within a reasonable time even with the most rough threshold 0.5. Whereas Bernstein-analysis succeeds with all values of the threshold and the degree. The number of stochastic classes is basically insensitive to both the parameters, ranging within 576 and 554, while computation time varies between less than 15 seconds and less than 5 minutes when the degree ranges from 2 to 6. Results of *Bernstein-analysis* can be compared only against those of the simulation (200,000 firings obtained in about 10 hours). With threshold 0.05 and degree 4, the difference has an average of 0.00037 and a maximum of 0.00268. With other degrees, the difference remains in any case under an average value of 0.00180 and a maximum of 0.01065. Besides, the threshold δ has no significant effect on the difference with respect to simulation results.

6.3 Fail&Repeat-All Model

Fig. 6 reports the *Fail&Repeat-All* model, which extends Fail&Repeat by allowing every resource to fail during the usage by any task. As in the previous model, failures occur in exponential time with rate 0.003, while repair is uniformly distributed in [200, 400].

Nondeterministic analysis enumerates 10,352 state classes in less than 10 seconds. Stochastic expansion is theoretically bounded as no persistent transitions exist after any firing of restart. However, as for the simpler Fail&Repeat, both *exact-analysis* and *relaxed-analysis* do not complete in a reasonable time even with the most rough threshold 0.5, whereas *Bernstein-analysis* completes the enumeration of the stochastic class graph with all the values of the threshold and the degree. The number of classes ranges between 5,449 and 5,749, which is lower than the number of nondeterministic classes due to the suppression of null probability transitions. Enumeration time is substantially insensitive to the threshold and increases with the degree, ranging from less than 16 minutes for degree 2 up to to a little more than 11 hours for degree 6.

To further stress the complexity of the analysis, Fail&Repeat-All can be further extended by replacing

Authorized licensed use limited to: Universita degli Studi di Firenze. Downloaded on July 2, 2009 at 03:23 from IEEE Xplore. Restrictions apply.

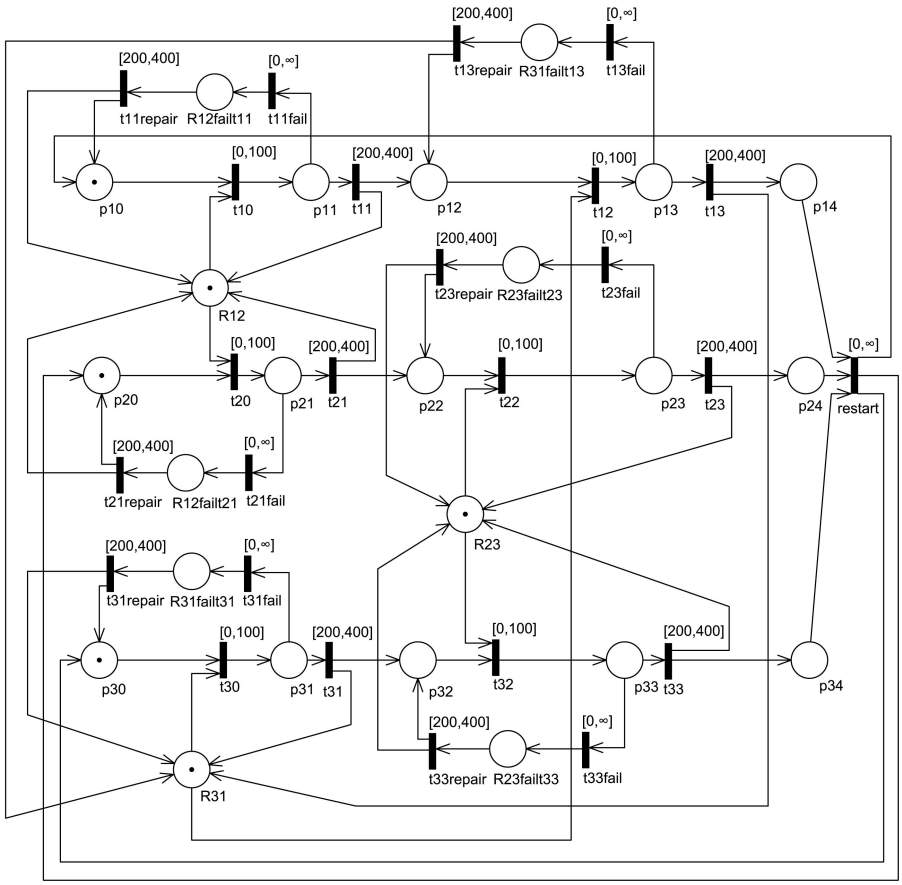


Fig. 6. (Fail&Repeat-All) A variant of Fail&Repeat where all operations using resources can fail. Failed actions are repeated after repair.

uniform distributions with the expolynomial functions $f_{wait}()$ and $f_{op}()$ that we used in the similar extension of Three-Tasks. Transition restart and resource failures are maintained exponential, while repair actions are distributed according to $f_{op}()$. The complexity increases, but *Bernstein-analysis* still completes the enumeration. The number of stochastic state classes remains basically invariant (between 5,715 and 5,670). Enumeration time does not depend on the threshold and increases with the degree, ranging from a little more than 45 minutes up to less than 29 hours for degrees between 2 and 5 (degree 6 was not tested being expected to complete in 3*29 hours).

6.4 Fail&Go Model

Fig. 7 shows a last variant of Fail&Repeat that we call *Fail&Go*: after the failure of a resource, the failed action is not repeated and the task continues the execution without waiting for the repair action. *Nondeterministic* analysis enumerates 1,424 state classes in less than 1 second. However, stochastic expansion is not guaranteed to be bounded: at the firing of restart, the repair action of one or both the resources used by the mid task may be still pending; a class reached through the firing of restart is no more guaranteed to be regenerative; and the stochastic class graph may thus include nonregenerative cycles [13].

Also the *Relaxed-analysis* does not terminate in a reasonable time even with threshold 0.5. *Bernstein-analysis* terminates with all thresholds and degrees, showing its capability to cope with the problem of theoretical unboundedness of the stochastic class graph. The number of stochastic state classes does not show significant variations, ranging between 947 and 975. The time of enumeration is not affected by the threshold and it increases with the degree, from less than 40 seconds up to less than 16 minutes with degrees ranging from 2 to 6. The comparison of the results of *Bernstein-analysis* with threshold 0.05 and degree 4 against the results of simulation (200,000 firings) shows an average difference equal to 0.00111 and a maximum equal to 0.00967. Variations on the threshold do not affect the difference.

7 Conclusions

The theory of stochastic state classes integrates the verification of feasible behaviors and the evaluation of their probability in the analysis of models with any number of concurrent generally distributed transitions. As a core step, the analysis process involves the derivation of state-density functions providing a measure of probability over the continuous set of states included in each nondeterministic state class. This faces practical and theoretical challenges, related to the complexity of symbolic manipulation of density functions over DBM domains and the exacerbation of state explosion during stochastic expansion of state classes, which may even be nonterminating.

In this paper, we developed a closed-form symbolic calculus for the derivation of stochastic classes, which takes advantage of efficient manipulation of DBM do-

Authorized licensed use limited to: Universita degli Studi di Firenze. Downloaded on July 2, 2009 at 03:23 from IEEE Xplore. Restrictions apply.

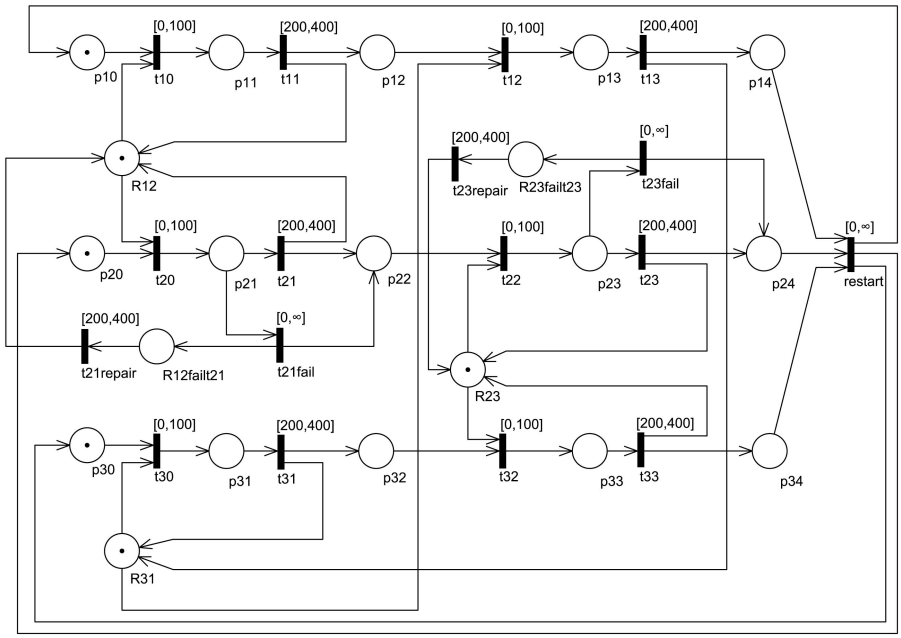


Fig. 7. (Fail&Go) A variant of Fail&Repeat that allows the mid task to continue the execution after a failure without waiting for the repair action.

mains and outperforms by various orders of magnitude a previous implementation based on the Mathematica toolbox [33]. The calculus also provides insight into analytic characteristics of the stochastic process underlying the model and points out its actual factors of complexity. These appear to be primarily related to the incremental partitioning of DBM domains in the piecewise representation of state-density functions. Based on this understanding, we developed an approach of approximate representation of the state space, which combines relaxation of the metrics of distance between density functions with approximation of state-density functions through Bernstein polynomials. The approach was developed for models including transitions with exponential distribution and transitions with expolynomial distribution over a finite support, but it can also be easily extended to encompass immediate and deterministic transitions in the light of the theory of [13].

In a practical perspective, computational experience indicates that the combined usage of the two approximation mechanisms substantially enlarges the range of applicability of the analysis with respect to the exact implementation of the closed-form calculus: on the one hand, distance relaxation limits the number of state-density functions that may be defined over the same state class, also in cases where this number might be infinite; on the other hand, Bernstein approximation provides a global analytic representation over the entire DBM domain, and thus maintains the cost of derivation of a successor class invariant with respect to memory acquired along nonregenerative behaviors. This enables completion of the enumeration for all tested models, some of which are, in fact, relatively complex and/or nonregenerative. Detailed analysis of results suggests that pushing further the limits of applicability of the approach is a matter of numerical precision much more than time and space.

In a more abstract perspective, and perhaps more importantly, the proposed technique comprises a novel approach to the analysis of non-Markovian models, which does not rely on the approximation of monovariate distributions in the model structure but rather on the approximation of multivariate joint probability functions in the model state space. This also seems a promising approach to attack extension toward models that break the structure of DBM encoding, and notably toward preemptive models [10], [14] that take major relevance in the context of real-time systems.

APPENDIX: THEOREM PROOFS

Lemma 4.1. *Integration bounds $E_o^+(x)$ and $L_o^+(x)$ are piecewise continuous functions with analytical representation within each element of a partition of D^b made of DBM subdomains R_{ij} with $i, j \in [0, N-1]$. Within each subdomain, R_{ij}, $E_o^+()$ and $L_o^+()$ are either constants or monovariate functions of a single component of the vector x*

$$L_o^+(x) = \begin{cases} B_{o*} & \text{iff } i = 0, \\ B_{i*} - x_i & \text{iff } i > 0, \end{cases}$$
$$E_o^+(x) = \begin{cases} -B_{*o} & \text{iff } j = 0 \\ -B_{*j} - x_j, & \text{iff } j > 0. \end{cases} \tag{41}$$

Proof. Let $\{\bar{R}_i\}_{i=0,N-1}$ be subsets of D^b defined as

$$\begin{aligned} \bar{R}_o &\stackrel{def}{=} D^b \bigcap \{x | x_i \le B_{i*} - B_{o*} \quad \forall i \in [1, N-1]\}, \\ \bar{R}_i &\stackrel{def}{=} D^b \bigcap \{x | x_i \ge B_{i*} - B_{o*} \\ &\qquad \wedge x_i - x_j \ge B_{i*} - B_{j*} \quad \forall j \in [1, N-1]\}. \end{aligned} \tag{42}$$

Authorized licensed use limited to: Universita degli Studi di Firenze. Downloaded on July 2, 2009 at 03:23 from IEEE Xplore. Restrictions apply.

According to (13a)

$$L_o^+(x) = \begin{cases} B_{o*} & \text{iff } x \in \bar{R}_o \\ B_{i*} - x_i & \text{iff } x \in \bar{R}_i. \end{cases} \quad (43)$$

Regions $\bar{R}_i$ are DBM zones being the restriction of a DBM through DBM constraints, their internal parts are disjoint by construction, their union can be proven to cover the entire domain D^b, and within each of them, $E_o^+(x)$ is either a constant (B_{o*}, in $\bar{R}_o$) or a monovariate linear function ($B_{i*} - x_i$, in each $\bar{R}_i$). In a similar manner, (13b) partitions the domain D^b into a finite set of subdomains $\{\underline{R}_i\}_{i=0,N-1}$ within each of which $E_o^+(x)$ is analytic. The intersection of the two partitions defines a set of DBM regions $\{R_{ij}\}_{i,j\in[0,N-1]}$ within each of which both $E_o^+(x)$ and $L_o^+(x)$ are analytic functions.

Since $E_o^+()$ and $L_o^+()$ are analytic within each subdomain R_{ij}, to prove that they are globally continuous functions, it will be sufficient showing that they are continuous along subdomain borders. For a point $\hat{x}$ at the border between two regions, say $\bar{R}_i$ and $\bar{R}_j$, (42) implies that

$$\begin{aligned} &\hat{x}_i - \hat{x}_j = B_{i*} - B_{j*}, \\ &\lim_{x\to\hat{x}, x\in\bar{R}_i} L_o^+(x) = \lim_{x\to\hat{x}} B_{i*} - x_i = B_{i*} - \hat{x}_i, \\ &\lim_{x\to\hat{x}, x\in\bar{R}_j} L_o^+(x) = \lim_{x\to\hat{x}} B_{j*} - x_j = B_{j*} - \hat{x}_j. \end{aligned} \quad (44)$$

The same steps can be applied to show that $E_o^+()$ also is continuous in $\hat{x}$. □

Theorem 4.1. *The state-density function is continuous and accepts a piecewise representation over a set of subdomains in DBM form, provided that the property is satisfied in the initial class and static density functions associated with transitions in the model are continuous functions defined by piecewise composition of analytic functions over a finite partition of the static firing interval.*

Proof. By induction, we assume that the property is satisfied in a class Σ, and we verify that it is maintained through the four steps of the derivation of the successor Σ' reached from Σ through t_o.

In the *conditioning* step, each subdomain is restricted by the same constraints that restrict D into D_{t_o}. Since the additional constraints are in DBM form, each subdomain (if not emptied) is maintained in DBM form. Of course, the property of continuity is maintained in the division by the constant factor μ_{t_o}.

In *time-advancement*, in the integration of (8), according to Lemma 4.1, $D_o^+(x)$ is an interval $D_o^+(x) = [E_o^+(x), L_o^+(x)]$. If $\phi_{\tau^a}()$ denotes the integral function of $f_{\tau^a}()$, we can thus express $f_{\tau^b}()$ as

$$\begin{aligned} f_{\tau^b}(x) &= \int_{E_o^+(x)}^{L_o^+(x)} f_{\tau^a}(x_o, x + x_o)dx_o \\ &= \phi_{\tau^a}(L_o^+(x), x + L_o^+(x)) - \phi_{\tau^a}(E_o^+(x), x + E_o^+(x)). \end{aligned} \quad (45)$$

Equation (45) directly implies that $f_{\tau^b}()$ is a continuous function. In fact, $\phi_{\tau^a}()$ is continuous, being the integral function of a continuous function; the difference and the composition of continuous functions is a continuous function; and finally, both $E_o^+(x)$ and $L_o^+(x)$ are continuous functions according to Lemma 4.1. As to the property of accepting an analytical representation, the domain of $f_{\tau^b}()$ must be partitioned twice so as to guarantee that both the integral function and the integration bounds accept an analytical representation. On the one hand, according to Lemma 4.1, $E_o^+(x)$ and $L_o^+(x)$ accept an analytical representation within a partition of D^a in DBM subdomains. On the other hand, the operations of time shift and projection of (8) transform each initial subdomain of $f_{\tau^a}()$ into a subdomain for $\phi_{\tau^a}(L_o^+(x), x + L_o^+(x)) - \phi_{\tau^a}(E_o^+(x), x + E_o^+(x))$. These subdomains are also in DBM form according to the basic property of closure of DBM encoding with respect to the succession relation between state classes [19], [37].

In the step of *disabling*, a projection is performed for each disabled transition. At each projection, continuity is preserved and the domain is subject to a partition refinement in the same way as that occurring for time-advancement.

Finally, in the step of *newly enabling*, the property of continuity is maintained due to the assumption that static density functions of transitions are continuous. No further partition refinement occurs if density functions are analytic over their support. If they are defined as piecewise functions, subdomains are not partitioned along the axis of times to fire of newly enabled transitions. □

Lemma 5.1. *if $EFT^s(t_i) = [0, \infty]$ and $f^s_{t_e} x_e = \lambda_e e^{-\lambda_e x_e}$, then in every stochastic class Σ where t_e is enabled, the firing domain and the state-density function can be expressed as*

$$\begin{aligned} D_{\langle\tau_e,\tau\rangle} &= \begin{cases} D_\tau \\ \tau_e - \tau_* \geq 0, \end{cases} \\ f_{\langle\tau_e,\tau\rangle}(x_e, x) &= \lambda_e e^{-\lambda_e x_e} f_\tau(x), \end{aligned} \quad (46)$$

where $\langle\tau_e, \tau\rangle$ denotes the vector of times to fire of enabled transitions and D_τ is the projection of $D_{\langle\tau_e,\tau\rangle}$ on the subspace of τ.

Proof. Equation (46) holds in every class where t_e is newly enabled as a direct consequence of the semantics of sTPNs. To prove the assertion by induction, we thus assume that (46) holds in Σ and show that it holds in any class Σ' that is reached from Σ through the firing of a transition t_o with t_e persistent. In the step of *conditioning*, τ_e^a becomes subject to the additional constraint $\tau_e^a - \tau_o^a \geq 0$, which is stronger than $\tau_e^a \geq 0$. The domain thus becomes

$$D_{\langle\tau_e^a,\tau^a\rangle} = \begin{cases} D_{\tau^a} \\ \tau_e^a - \tau_o^a \geq 0. \end{cases} \quad (47)$$

In the step of time-advancement, τ_e^a is replaced by $\tau_e^b = \tau_e^a - \tau_o^a$ and the domain is rewritten as

$$D_{\langle\tau_e^b,\tau^b\rangle} = \begin{cases} D_{\tau^b} \\ \tau_e^b \geq 0. \end{cases} \quad (48)$$

The subsequent steps of disabling and newly enabling do not affect the time to fire of t_e, and thus, the final domain of class Σ' is written as

Authorized licensed use limited to: Universita degli Studi di Firenze. Downloaded on July 2, 2009 at 03:23 from IEEE Xplore. Restrictions apply.

$$D_{\langle \tau'_e, \tau' \rangle} = \begin{cases} D_{\tau'} \\ \tau'_e \geq 0. \end{cases} \quad (49)$$

Besides, for the state-density function, the form of (46) is maintained in the division of the conditioning step. In time-advancement

$$f_{\langle \tau^b_e, \tau^b \rangle}(x_e, x) = \lambda_e e^{-\lambda_e x_e} \int_{E^+_o(x_e,x)}^{L^+_o(x_e,x)} \frac{1}{\mu_{t_o}} e^{-\lambda_e x_o} f_\tau(x + x_o) dx_o. \quad (50)$$

Both $E^+_o(x_e, x)$ and $L^+_o(x_e, x)$ do not actually depend on x_e. In fact, τ_e does not determine any constraint for any of the variables in τ and in particular not for τ_o. The integral in the right part of the equation thus does not depend on x_e and the entire expression can be rewritten as

$$f_{\langle \tau^b_e, \tau^b \rangle}(x_e, x) = \lambda_e e^{-\lambda_e x_e} f_{\tau^b}(x). \quad (51)$$

In a similar manner, in the step of disabling, the form of () is maintained as a consequence of the fact that integration bounds $D^+_o(x_e, x)$ and $D^+_o(x_e, x)$ do not actually depend on x_e. The step of newly enabling does not change the form as the times to fire of newly enabled transitions are added in the form of a product as independent variables. □

Theorem 5.1. *In every reachable stochastic state class $\Sigma = \langle m, D, f_\tau() \rangle$, the firing domain D and the state-density function $f_\tau()$ can be expressed as*

$$f_{\langle \alpha,\beta,\gamma \rangle}(x, y, z) = f_\alpha(x) \cdot \prod_{n=0}^{N-1} f^s_{t_{\nu(n)}}(y_n) \cdot \prod_{e=0}^{E-1} \lambda_\epsilon(e) e^{-\lambda_{\epsilon(e)} z_{\epsilon(e)}},$$

$$D_{\langle \alpha,\beta,\gamma \rangle} = D_\alpha(x) \times \left(\prod_{n=0}^{N-1} [EFT^s(t_{\nu(n)}), LFT^s(t_{\nu(n)})] \right) \times [0, \infty]^E. \quad (52)$$

Proof. The statement is an immediate consequence of Lemma 5.1 and (10). □

Algorithm 8.1.
Let $x = \langle x_o, \ldots, x_{N-1} \rangle$ be a point in the minimum embedding rectangle R_α of D_α, b_{ij} denote the coefficients of the normal form of D_α, and x^k denote x at the nth step of the algorithm

$x^o = x$;
for($k = 0, N-1$)
  if ($\max_j\{-b_{jk} + x^k_j\} \leq x^k_k \leq \min_j\{b_{kj} + x^k_j\}$)
    $x^{k+1}_k = x^k_k$;
  else if ($x^k_k < \max_j\{-b_{jk} + x^k_j\}$)
    $x^{k+1}_k = \max_j\{-b_{jk} + x^k_j\}$;
  else if ($x^k_k \geq \min_j\{b_{kj} + x^k_j\}$)
    $x^{k+1}_k = \min_j\{b_{kj} + x^k_j\}$.

Remark 8.1. (complexity): Algorithm 8.1 has complexity $O(N^2)$ and can be embedded in a test that verifies whether $x \in D_\alpha$ adding just the cost for no more than N assignments.

Lemma 8.1. *(correctness): If $D_\alpha \neq \phi$: if $x \in D_\alpha$, then $x^{N-1} = x$; whereas if $x \notin D_\alpha$, then x^{N-1} is a point on the border of D_α.*

Proof. To prove the statement, it is sufficient proving that at any step of the execution $\max_j\{-b_{jk} + x^k_j\} \leq \min_j\{b_{kj} + x^k_j\}$. Ab absurdo suppose that $\max_j\{-b_{jk} + x^k_j\} > \min_j\{b_{kj} + x^k_j\}$. This implies that there exists $l, m < k$ such that

$$-b_{lk} + x^k_l > b_{km} + x^k_m. \quad (53)$$

If $l = m$, (53) implies $0 > b_{kl} + b_{lk}$, which, in turn, implies that $D_\alpha = \phi$.

If $l > m$, then by construction, $x^k_l \leq \min_j\{b_{lj} + x^k_j\} \leq b_{lm} + x^k_m$. By replacing in (53), this implies $-b_{lk} + b_{lm} + x^k_m > -b_{lk} + x^k_l > b_{km} + x^k_m$, and thus $b_{lm} > b_{lk} + b_{km}$, which is not consistent with the assumption that coefficients b_{ij} are in normal form. The case $m > l$ is equivalent to the case $l > m$. □

Lemma 8.2. *(continuity): The algorithm maps close points into close points.*

Proof. We just sketch the proof, which is conceptually simple but technically tedious.

We assume that x and $\bar{x}$ are close points (i.e., $\|\bar{x} - x\| \leq \epsilon$). By induction, we assume that after k steps, the points are still close and we show that they remain close after the $(k+1)$th step. To this end, it is sufficient noting that in the kth step, only the kth component of one or both points can be changed. If the two points are moved along different direction, then the distance between the cuts along these directions can be bounded by the distance between the points themselves. In the end, we obtain an estimate on the distance between $\|x^{N-1} - \bar{x}^{N-1}\|$, which runs to zero with ϵ. □

ACKNOWLEDGMENTS

This work was partially supported by the SW Initiative of the Finmeccanica Group under research line "Design and Verification Methods for Concurrent Real Time SW."

REFERENCES

[1] M.M. Ajmone, G. Balbo, and G. Conte, "A Class of Generalized Stochastic Petri Nets for the Performance Evaluation of Multiprocessor Systems," *ACM Trans. Computer Systems*, 1984.
[2] R. Alur, C. Courcoubetis, and D. Dill, "Model-Checking for Real-Time Systems," *Proc. Fifth Symp. Logic in Computer Science*, June 1990.
[3] R. Alur and D.L. Dill, "Automata for Modeling Real-Time Systems," *Proc. 17th Int'l Colloquium Automata, Languages and, Programming*, 1990.
[4] G. Behrmann, A. David, and K.G. Larsen, "A Tutorial on UPPAAL," *Formal Methods for the Design of Real-Time Systems: Proc. Fourth Int'l School Formal Methods for the Design of Computer, Comm., and SW Systems*, pp. 200-236, Sept. 2004.
[5] G. Behrmann, K.G. Larsen, J. Pearson, C. Weise, and W. Yi, "Efficient Timed Reachability Analysis Using Clock Difference Diagrams," *Proc. 11th Int'l Conf. Computer Aided Verification*, 1999.
[6] B. Berthomieu and M. Diaz, "Modeling and Verification of Time Dependent Systems Using Time Petri Nets," *IEEE Trans. Software Eng.*, vol. 17, no. 3, pp. 259-273, Mar. 1991.
[7] B. Berthomieu and M. Menasche, "An Enumerative Approach for Analyzing Time Petri Nets," *Information Processing: Proc. IFIP Congress 1983*, R.E.A. Mason, ed., vol. 9, pp. 41-46, 1983.

Authorized licensed use limited to: Universita degli Studi di Firenze. Downloaded on July 2, 2009 at 03:23 from IEEE Xplore. Restrictions apply.

[8] B. Berthomieu, P.-O. Ribet, and F. Vernadat, "The Tool TINA—Construction of Abstract State Spaces for Petri Nets and Time Petri Nets," *Int'l J. Production Research,* vol. 42, no. 14, 2004.
[9] A. Bobbio and M. Telek, "A Benchmark for PH Estimation Algorithms: Result for Acyclic-PH," *Stochastic Models,* vol. 10, pp. 661-677, 1994.
[10] G. Bucci, A. Fedeli, L. Sassoli, and E. Vicario, "Timed State Space Analysis of Real Time Preemptive Systems," *IEEE Trans. Software Eng.,* vol. 30, no. 2, pp. 97-111, Feb. 2004.
[11] G. Bucci, R. Piovosi, L. Sassoli, and E. Vicario, "Introducing Probability within State Class Analysis of Dense Time Dependent Systems," *Proc. Second Int'l Conf. Quantum Evaluation of Systems,* Sept. 2005.
[12] G. Bucci, L. Sassoli, and E. Vicario, "Oris: A Tool for State Space Analysis of Real-Time Preemptive Systems," *Proc. First Int'l Conf. Quantum Evaluation of Systems,* 2004.
[13] L. Carnevali, L. Sassoli, and E. Vicario, "Using Stochastic State Classes in Quantitative Evaluation of Dense-Time Reactive Systems," *IEEE Trans. Software Eng.,* accepted for publication.
[14] F. Cassez and K. Larsen, *The Impressive Power of Stopwatches,* Aug. 2000.
[15] H. Choi, V.G. Kulkarni, and K. Trivedi, "Markov Regenerative Stochastic Petri Nets." *Performance Evaluation,* vol. 20, pp. 337-357, 1994.
[16] G. Ciardo, R. German, and C. Lindemann, "A Characterization of the Stochastic Process Underlying a Stochastic Petri Net," *IEEE Trans. Software Eng.,* vol. 20, no. 7, pp. 506-515, July 1994.
[17] G. Ciardo, J. Muppala, and K.S. Trivedi, "SPNP: Stochastic Petri Net Package," *IEEE Int'l Workshop Petri Nets and Performance Models,* pp. 142-151, 1989.
[18] C. Daws, A. Olivero, S. Tripakis, and S. Yovine, "The Tool KRONOS," *Proc. Hybrid Systems III,* 1995.
[19] D. Dill, "Timing Assumptions and Verification of Finite-State Concurrent Systems," *Proc. Workshop Computer Aided Verification Methods for Finite State Systems,* 1989.
[20] A. Feldmann and W. Whitt, "Fitting Mixtures of Exponentials to Long-Tail Distributions to Analyze Network Performance Models," *Performance Evaluation,* vol. 31, nos. 3-4, pp. 245-279, 1998.
[21] G. Gardey, D. Lime, M. Magnin, and O. Roux, "Romeo: A Tool for Analyzing Time Petri Nets," *Proc. 17th Int'l Conf. Computer Aided Verification,* 2005.
[22] R. German, "Transient Analysis of Deterministic and Stochastic Petri Nets by the Method of Supplementary Variables," *Proc. Third Int'l Workshop Modeling, Analysis, and Simulation of Computer and Telecomm. Systems,* 1995.
[23] R. German and M. Telek, "Formal Relation of Markov Renewal Theory and Supplementary Variables in the Analysis of Stochastic Petri Nets," *Proc. Eighth Int'l Workshop Petri Nets and Performance Models,* pp. 64-73, 1999.
[24] C. Lindemann and G.S. Shedler, "Numerical Analysis of Deterministic and Stochastic Petri Nets with Concurrent Deterministic Transitions," *Performance Evaluation,* vols. 27/28, pp. 565-582, 1996.
[25] C. Lindemann and A. Thuemmler, "Transient Analysis of Deterministic and Stochastic Petri Nets with Concurrent Deterministic Transitions," *Performance Evaluation,* vols. 36/37, pp. 35-54, 1999.
[26] G. Lorentz, *Bernstein Polynomials.* Univ. of Toronto Press, 1953.
[27] S. Manolache, P. Eles, and Z. Peng, "Schedulability Analysis of Multiprocessor Real-Time Applications with Stochastic Task Execution Times," *Proc. 20th Int'l Conf. Computer Aided Design,* pp. 699-706, Nov. 2002.
[28] P. Merlin and D. Farber, "Recoverability of Communication Protocols," *IEEE Trans. Comm.,* vol. 24, no. 9, 1976.
[29] M.K. Molloy, "Discrete Time Stochastic Petri Nets," *IEEE Trans. Software Eng.,* vol. 11, no. 4, pp. 417-723, Apr. 1985.
[30] M.F. Neuts, *Matrix Geometric Solutions in Stochastic Models.* Johns Hopkins Univ. Press, 1981.
[31] W. Penczek and A. Polrola, "Specification and Model Checking of Temporal Properties in Time Petri Nets and Timed Automata," *Proc. 25th Int'l Conf. Application and Theory of Petri Nets,* June 2004.
[32] L. Sassoli and E. Vicario, "Analysis of Real Time Systems Through the Oris Tool," *Proc. Fourth Int'l Conf. Quantum Evaluation of Systems,* Sept. 2006.
[33] L. Sassoli and E. Vicario, "Close Form Derivation of State-Density Functions over DBM Domains in the Analysis of Non-Markovian Models," *Proc. Fifth Int'l Conf. Quantum Evaluation of Systems,* Sept. 2007.
[34] T. Sauer, "Multivariate Bernstein Polynomials and Convexity," *Computer Aided Geometric Design,* vol. 8, no. 6, pp. 465-478, 1991.
[35] M. Telek and A. Horvarth, "Supplementary Variable Approach Applied to the Transient Analysis of Age-MRSPNs," *Proc. Int'l Performance and Dependability Symp.,* pp. 44-51, 1998.
[36] E. Teruel, G. Franceschinis, and M. DePierro, "Well Defined Generalized Stochastic Petri Nets: A Net Level Method to Specify Priorities," *IEEE Trans. Software Eng.,* vol. 29, no. 11 pp. 962-973, Nov. 2003.
[37] E. Vicario, "Static Analysis and Dynamic Steering of Time Dependent Systems Using Time Petri Nets," *IEEE Trans. Software Eng.,* vol. 27, no. 8, pp. 728-748, Aug. 2001.
[38] WolframResearch, Mathemathica 5.2, www.wolfram.com, 2009.
[39] A. Zimmermann, J. Freiheit, R. German, and G. Hommel, "Petri Net Modelling and Performability Evaluation with TimeNET 3.0," *Proc. 11th Int'l Conf. Modelling Techniques and Tools for Computer Perfermonce Evaluation,* pp. 188-202, 2000.

Laura Carnevali received the doctoral degree in informatics engineering from the University of Florence in 2006, where she is currently working toward the PhD degree in informatics, multimedia, and telecommunications engineering. Her research interests mainly focus on correctness verification and performance evaluation of real-time reactive systems, with specific interest on stochastic characterization of timed models. Her activity also addresses the integration of formal methods in the development life cycle of real-time software. She is a member of the Software Technologies Lab at the University of Florence.

Leonardo Grassi received the MSc degree in informatics engineering from the University of Florence, Italy, in 2006. From 2006 to 2008, he was a member of the Software Technologies Lab at the University of Florence, where his work centered on methods for the design and testing of embedded real-time applications. He is currently working as a software engineer in industry. His research interests are in verification techniques of safety critical software, software development processes, and source code analysis.

Enrico Vicario received the doctoral degree in electronics engineering and the PhD degree in information and telecommunications engineering from the University of Florence in 1990 and 1994, respectively. He is a full professor of computer science in the School of Engineering at the University of Florence, where he teaches computer-programming, sofware engineering, and software verification-and-testing. His present research activity is mainly focused on state-space analysis of concurrent-timed systems for correctness verification, quantitative evaluation, and real-time testing. He also carries out technological experimentation and transfer in the area of software architectures and design. He is a member of the IEEE and the IEEE Computer Society.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.

Authorized licensed use limited to: Universita degli Studi di Firenze. Downloaded on July 2, 2009 at 03:23 from IEEE Xplore. Restrictions apply.