ELSEVIER

Performance Evaluation 36–37 (1999) 35–54

PERFORMANCE EVALUATION
An International Journal

www.elsevier.com/locate/peva

Transient analysis of deterministic and stochastic Petri nets with concurrent deterministic transitions

Christoph Lindemann *, Axel Thümmler [1]

University of Dortmund, Department of Computer Science, August-Schmidt-Str. 12, 44227 Dortmund, Germany

Abstract

This paper introduces an efficient numerical algorithm for transient analysis of deterministic and stochastic Petri nets (DSPNs) and other discrete-event stochastic systems with exponential and deterministic events. The proposed approach is based on the analysis of a general state space Markov chain (GSSMC) whose state equations constitute a system of multidimensional Fredholm integral equations. Key contributions of this paper constitute the observations that the transition kernel of this system of Fredholm equations is piece-wise continuous and separable. Due to the exploitation of these properties, the GSSMC approach shows great promise for being effectively applicable for the transient analysis of large DSPNs with concurrent deterministic transitions. Moreover, for DSPNs without concurrent deterministic transitions the proposed GSSMC approach requires three orders of magnitude less computational effort than the previously known approach based on the method of supplementary variables.

Keywords: Generalized semi-Markov processes; General state space Markov chains; Numerical solvers for Fredholm integral equations; Performance and dependability models for high-speed networks

1. Introduction

Deterministic and stochastic Petri nets (DSPNs) introduced by Ajmone Marsan and Chiola in [2] are a stochastic modeling formalism with graphical representation which include both exponentially distributed and deterministic delays. Under the restriction that in any marking of a DSPN at most one deterministic transition is enabled, a highly efficient numerical method for steady state analysis has been introduced [11] and implemented in a software package [12]. While steady state analysis allows the evaluation of long run behavior of computer and telecommunication systems, a considerable number of important performance and dependability studies require the analysis of time-dependent behavior; i.e., transient analysis.

Previous work on transient analysis of DSPNs was always based on the restriction that deterministic transitions are not concurrently enabled. Choi, Kulkarni, and Trivedi observed that the marking

* Corresponding author. E-mail: cl@cs.uni-dortmund.de
[1] E-mail: thummler@ls4.cs.uni-dortmund.de

PII: S0166-5316(99)00020-6

process underlying a DSPN with this restriction is a Markov regenerative stochastic process [4]. They introduced a numerical method for transient analysis of such DSPNs based on numerical inversion of Laplace-Stieltjes transforms. While this numerical method is certainly of theoretical interest, it is not suitable for transient analysis of large DSPNs. More recently, German et al. developed a numerical method for transient analysis of DSPNs based on the method of supplementary variables (see e.g., [5,10]). Using the same approach in a recent paper, Telek and Horvath developed state equations for transient analysis of Markov regenerative stochastic Petri nets in which timed transitions keep their remaining firing times in case their firing process gets preempted for subsequent resumption instead of discarding them and restarting the firing process [16]. Unfortunately, the practical applicability of the supplementary variables approach is severely limited because it requires, already in this restricted case, numerical solution of a system of partial differential equations.

This paper introduces an effective numerical method for transient analysis of deterministic and stochastic Petri nets (DSPNs) without structural restrictions on the enabling of deterministic transitions. The proposed approach is based on the analysis of a general state space Markov chain (GSSMC) whose state equations constitute a system of multidimensional Fredholm integral equations. Key contributions of this paper constitute the observations that the transition kernel of this system of Fredholm equations is piece-wise continuous and separable. It is known (see e.g., [3]) that numerical solution of such Fredholm equations requires considerably less computational cost than numerical solution of partial differential equations.

We illustrate that the GSSMC approach shows great promise for being effectively applicable for the transient analysis of quite large DSPN with deterministic transitions concurrently enabled. Furthermore, for DSPNs without concurrent deterministic transitions, the GSSMC approach is three orders of magnitude faster than the previously known method based on the supplementary variables approach [5,10]. In particular, for a DSPN as considered in [10] the GSSMC approach requires a couple of minutes whereas the refined implementation of the supplementary variables approach requires more than 100 hours of CPU time. This considerable gain in computational efficiency over the approach based on supplementary variables is due to the following reasons: In the GSSMC approach, the integration domain is decomposed in disjoint regions in which the elements of the transition kernel are *continuous and differentiable.* Thus, the GSSMC approach does not have to deal with discontinuities in functional expressions in the computational scheme. Furthermore, we observe that the transition kernel of the GSSMC is *separable*. That is the kernel matrix can be expressed as the sum of a matrix comprising only constant entries, a matrix comprising only functional entries setting new clocks (i.e., comprising only of functional entries in a_1 and/or a_2) and a matrix comprising only functional entries taking into account old clocks (i.e., comprising only of functional entries in c_1 and/or c_2).

Throughout this paper, we state the GSSMC approach in the context of DSPNs. However, the GSSMC approach is by no means restricted to analysis of DSPNs. The method introduced in this paper is also applicable for the transient analysis of queueing networks, stochastic process algebras, and other discrete-event stochastic systems with an underlying stochastic process which can be represented as a generalized semi-Markov process with exponential and deterministic events.

The remainder of this paper is organized as follows. In Section 2 we show how to define the general state space Markov chain underlying a DSPN with concurrent deterministic transitions and introduce the notation. Section 3 describes the form of the transition kernel. The iterative scheme on the system of state equations of the GSSMC is presented in Section 4. To illustrate the applicability of the GSSMC approach for transient analysis of DSPNs, in Section 5 we provide curves plotting computational effort and memory requirements for transient analysis of DSPNs of an MMPP/D/2/K queue and an MMPP/D/1/K queue with failure and repair as already considered in [10]. Finally, concluding remarks are given (Section 6).

2. Analysis of the marking process of a DSPN by a general state space Markov chain

Formally, a Petri net is a directed bipartite graph with one set of vertices called *places* (drawn as circles) and the other called *transitions* (drawn as bars). Places may contain *tokens* which are drawn as dots. Places and transitions are connected by directed arcs or inhibitor arcs (drawn with an circled head). Arcs may be labeled with integer numbers denoting their multiplicity. The default multiplicity of an arc is one. A transition is said to be *enabled*, if all of its input places contain at least as many tokens as the multiplicity of the corresponding input arc and all of its inhibitor places contain less tokens than the multiplicity of the corresponding inhibitor arc. A transition *fires* by removing from each input place as many tokens as the multiplicity of the corresponding input arc, and by adding to each output place as many tokens as the multiplicity of the corresponding output arc. In deterministic and stochastic Petri Nets (DSPNs [2]) three types of transitions exist: immediate transitions drawn as thin bars fire without delay, exponential transitions drawn as empty bars fire after an exponentially distributed delay whereas deterministic transitions drawn as black bars fire after a constant delay.

Numerical analysis of DSPNs proceeds by computing transient or stationary distributions for its underlying continuous-time stochastic process $\{S(t)\colon t \geq 0\}$. The process $\{S(t)\colon t \geq 0\}$ has a discrete state space (i.e., the tangible markings of the DSPN) and is denoted as the *marking process* of the DSPN. Since the deterministic distribution does not show absence of memory, a proper definition of the stochastic behavior of DSPNs requires the specification how the selection of the next transition to fire is performed and how the model keeps track on past history. Throughout this paper, we assume that among all enabled timed transitions in a DSPN the one with the minimum remaining firing time determines the next marking change. Furthermore, after a marking change each timed transition newly enabled samples a remaining firing time from its firing delay distribution and each timed transition, which has already been enabled in the previous marking and is still enabled in the current marking, keep its remaining firing time. This stochastic behavior corresponds to the execution policy *race with enabling memory* as defined in [1]. We assume that the reachability graph of the DSPN comprises of a finite number of tangible markings, denoted by N.

In [13], we showed that the marking process $\{S(t)\colon t \geq 0\}$ can be represented as a generalized semi-Markov process with exponential and deterministic events. The execution policy race with enabling memory also coincidences with the usual state transition mechanism in a generalized semi-Markov process (see e.g., [7,8]). For stationary analysis of the continuous-time marking process underlying a DSPN with concurrent deterministic transitions, we defined in [13] a discrete-time general state space Markov chain (GSSMC). A GSSMC is completely specified by a transition kernel (heuristically, this is a family of probability matrices) and an initial distribution at time $t = 0$. As shown in [13], given the tangible reachability graph of a DSPN, the transition kernel of the underlying GSSMC can be numerically determined by extending the concept of subordinated Markov chains introduced in [11]. As explained below, the initial distribution of the GSSMC can easily be derived from the initial marking of the corresponding DSPN.

We enumerate the deterministic transitions of the DSPN by $t_1, t_2, \ldots, t_M$ and define D_m to be the firing delay of transition t_m ($1 \le m \le M$). Let $C_m(t)$ be the remaining firing time associated with deterministic transition t_m at time $t \ge 0$. In any state in which deterministic transition t_m is not enabled, we set $C_m(t) = 0$. To derive the GSSMC underlying a DSPN, we define a discrete-time process $\{X(nD)\colon n \ge 0\}$ by observing the marking process $\{S(t)\colon t \ge 0\}$ at a sequence $\{nD\colon n \ge 0\}$ of fixed times for some appropriately defined step size $D > 0$.

$$X(nD) = (S(nD), C_1(nD), C_2(nD), \ldots, C_M(nD)) . \tag{1}$$

The process $S(nD)$ represents the state (i.e., tangible marking of the DSPN) and $C_m(nD)$ represents the mth component of the clock-reading vector (i.e., remaining firing time of deterministic transitions t_m) at instant of time nD. The memoryless property of the exponential distribution implies that $\{X(nD)\colon n \ge 0\}$ is a GSSMC, i.e., it satisfies the Markov property.

For ease of exposition, we restrict the discussion in this paper to DSPNs in which at most two deterministic transitions may be concurrently enabled. Then, the subset of tangible markings (i.e., states of the GSSMC) in which only exponential transitions are enabled is denoted by S_{exp}. Similarly, the subsets of states in which one deterministic transition and two deterministic transitions are (concurrently) enabled are denoted by S_{det1} and S_{det2}, respectively. Without loss of generality, we enumerate the states of the marking process as follows:

$$\begin{aligned} S_{\text{exp}} &= \{s_1, s_2, \ldots, s_{N_1}\}, \\ S_{\text{det1}} &= \{s_{N_1+1}, s_{N_1+2}, \ldots, s_{N_1+N_2}\}, \\ S_{\text{det2}} &= \{s_{N_1+N_2+1}, s_{N_1+N_2+2}, \ldots, s_N\}. \end{aligned} \tag{2}$$

The derivation of the initial distribution of the GSSMC, denoted by X_0, from initial (tangible) marking of the DSPN is straight forward. To specify X_0, we define initial state probabilities at time $t = 0$ as:

$$\begin{aligned} &\pi_i^0 = P\{S(0) = s_i\} && \text{for } 1 \le i \le N_1, \\ &\pi_i^0(a_1) = P\{S(0) = s_i, C_{l(i)}(0) \le a_1\} && \text{for } N_1 + 1 \le i \le N_1 + N_2, \\ &\pi_i^0(a_1, a_2) = P\{S(0) = s_i, C_{l(i)}(0) \le a_1, C_{m(i)}(0) \le a_2\} && \text{for } N_1 + N_2 + 1 \le i \le N. \end{aligned} \tag{3}$$

In (3), we denote the index of the deterministic transition(s) enabled in a state s_i by $l(i)$ and $m(i)$, respectively and neglect other zero-valued remaining firing times. If the marking process of the DSPN resides at time 0 in some state $s_i \in S_{\text{det1}}$ and deterministic transition $t_{l(i)}$ just starts its firing procedure, we obtain the following initial distribution:

$$\begin{aligned} &\pi_j^0 = 0 && \text{for } 1 \le j \le N_1, \\ &\pi_i^0(D_{l(i)}) = 1.0,\ \pi_j^0(D_{l(j)}) = 0 && \text{for } N_1 + 1 \le j \le N_1 + N_2 \text{ with } j \ne i, \\ &\pi_j^0(D_{l(j)}, D_{m(j)}) = 0 && \text{for } N_1 + N_2 + 1 \le j \le N. \end{aligned}$$

If the initial marking $s_i \in S_{\text{exp}}$ or $s_i \in S_{\text{det2}}$, the initial distribution of the underlying GSSMC is derived accordingly. Note that the GSSMC approach allows the consideration of more general initial distributions than those that are arising from the assumption that the DSPN resides initially in a single tangible marking in which deterministic transitions just start their firing procedure. In general, any initial distribution subject to the normalization condition (see Eq. (21) in Section 4) is a feasible initial state.

Given the initial distribution X_0 for the GSSMC and using (2), we define for the GSSMC underlying a DSPN with two deterministic transitions concurrently enabled three kinds of time-dependent state probabilities:

$$\begin{aligned} \pi_i^{(n)} &= P\{S(nD) = s_i \mid X_0\} && \text{for } s_i \in S_{\text{exp}}, \\ \pi_i^{(n)}(a_1) &= P\{S(nD) = s_i, C_{l(i)}(nD) \le a_1 \mid X_0\} && \text{for } s_i \in S_{\text{det1}}, \\ \pi_i^{(n)}(a_1, a_2) &= P\{S(nD) = s_i, C_{l(i)}(nD) \le a_1, C_{m(i)}(nD) \le a_2 \mid X_0\} && \text{for } s_i \in S_{\text{det2}} \\ &&& \text{for } n = 1, 2, \ldots \text{ and } 0 < a_1, a_2 \le D. \end{aligned} \tag{4}$$

Subsequently, the transient state probabilities of the marking process of the DSPN at instants of time $t = nD$ are given by $\pi_i^{(n)}$ for $s_i \in S_{\text{exp}}$, $\pi_i^{(n)}(D_{l(i)})$ for $s_i \in S_{\text{det1}}$ and, $\pi_i^{(n)}(D_{l(i)}, D_{m(i)})$ for $s_i \in S_{\text{det2}}$, respectively.

To determine the transient state probabilities of the marking process of a DSPN at an arbitrary instant of time $t > 0$, we determine the number of iterations in the scheme (24) to (28), $n_0 = \lfloor t/D \rfloor$, and the remaining instant of time, $a = t - \lfloor t/D \rfloor$. Then, we derive the transient state probabilities at time t from the state probability vector at time $n_0 D$ by a forward Chapman-Kolmogorov equation of the GSSMC with step size a. By this approach, we exploit that due to the definition of the GSSMC holds for $0 < a \le D$:

$$\begin{aligned} \pi_i^{(n_0D+a)} &= P\{S(n_0D + a) = s_i \mid X_0\} && \text{for } s_i \in S_{\text{exp}}, \\ \pi_i^{(n_0D+a)}(D_{l(i)}) &= P\{S(n_0D + a) = s_i, C_{l(i)}(n_0D + a) \le a \mid X_0\} && \text{for } s_i \in S_{\text{det1}}, \\ \pi_i^{(n_0D+a)}(D_{l(i)}, D_{m(i)}) & \\ = P\{S(n_0D + a) = s_i, C_{l(i)}&(n_0D + a) \le a, C_{m(i)}(n_0D + a) \le a \mid X_0\} && \text{for } s_i \in S_{\text{det2}}. \end{aligned} \tag{5}$$

In [15], a similar decomposition of the transient instant of time has been employed for computing dependability measures of a particular single server queueing systems with deterministic repair.

Furthermore, as shown in [13], the stationary or time-averaged distribution of the discrete-time process $\{S(nD): n \ge 0\}$ is equal to the stationary or time-averaged distribution of the continuous-time process $\{S(t): t \ge 0\}$. That is:

$$\begin{aligned} \lim_{t\to\infty} P\{S(t) = s_i\} &= \lim_{n\to\infty} P\{S(nD) = s_i\} && \text{if a stationary solution exists,} \\ \lim_{t\to\infty} \frac{1}{t}\int_0^t P\{S(u) = s_i\}\,\mathrm{d}u &= \lim_{n\to\infty} \frac{1}{n}\sum_{k=0}^{n} P\{S(kD) = s_i\} && \text{otherwise} \\ &&& \text{for } 1 \le i \le N. \end{aligned} \tag{6}$$

Note that if the GSSMC $\{X(nD): n \ge 0\}$ is not only aperiodic and positive recurrent but also has a regeneration set, both this discrete-time process and the continuous-time marking process of the corresponding DSPN have a stationary distribution. In the examples considered in Section 5, a stationary distribution exists, only if the effective arrival rates are smaller than the service rates (e.g., $\lambda_{\text{eff}} < 2/D$ for the MMPP/D/2/K queue). Otherwise, the long run behavior of the marking process can be evaluated by the time-averaged distribution of the corresponding GSSMC.

3. The transition kernel

The transition kernel of the GSSMC specifies one-step jump probabilities from a given state at instant of time nD to all reachable new states at instant of time $(n+1)D$. As for an ordinary discrete-time Markov chain, for all states s_j not reachable from s_i corresponding jump probabilities $p_{ij}(\cdot)$ are set to zero. In general, entries of the transition kernel of a GSSMC are functions of clock readings c_1 and c_2 associated with the current state s_i and intervals for clock readings $(0, a_1]$ and $(0, a_2]$ associated with the new state s_j. In this section, we derive the form of the transition kernel of the GSSMC underlying a DSPN with at most two deterministic transitions concurrently enabled. Without loss of generality, we always assume $a_1 \leq a_2$.

Given the GSSMC $\{X(nD)\colon n \geq 0\}$ underlying a DSPN with at most two concurrently enabled deterministic transitions, the transition kernel can be expressed by a functional matrix $\boldsymbol{P}(c_1, c_2, a_1, a_2)$. For state transitions within the set S_{det2} elements of $\boldsymbol{P}(c_1, c_2, a_1, a_2)$ are defined as conditional probabilities that the next state is s_j with clock readings $C_{l(j)} \in (0, a_1]$ and $C_{m(j)} \in (0, a_2]$ given that the current state is s_i with clock readings $C_{l(i)} = c_1$ and $C_{m(i)} = c_2$. That is:

$$p_{ij}(c_1, c_2, a_1, a_2) = P\{S((n+1)D) = s_j, C_{m(j)}((n+1)D) \leq a_1, C_{l(j)}((n+1)D) \leq a_2 \\ \mid S(nD) = s_i, C_{m(i)}(nD) = c_1, C_{l(i)}(nD) = c_2\}. \tag{7}$$

Omitting the second clock reading, we obtain the general form of transition probabilities from S_{det2} to S_{det1} and within S_{det1}. Elements of $\boldsymbol{P}(c_1, c_2, a_1, a_2)$ representing transition probabilities within S_{exp} are defined as conditional probabilities that the next state is s_j given that the current state is s_i; i.e., as in an ordinary DTMC. That is:

$$p_{ij}(D) = P\{S((n+1)D) = s_j \mid S(nD) = s_i\}. \tag{8}$$

To determine transient state probabilities of the GSSMC at arbitrary instants of time (i.e., not only at $t = nD$ where $n = 1, 2, \ldots$), kernel elements of the form (8) are needed for all $0 < a_1 \leq D$. Thus in general, elements of $\boldsymbol{P}(c_1, c_2, a_1, a_2)$ representing transition probabilities within S_{exp} are defined as:

$$p_{ij}(a_1) = P\{S(nD + a_1) = s_j \mid S(nD) = s_i\} \quad \text{for } 0 < a_1 \leq D. \tag{9}$$

Transition probabilities from S_{det2} and S_{det1} to S_{exp} are defined similar to (9). Because two deterministic transitions may be enabled concurrently, we have to consider several cases for the clock readings c_1 and c_2. In general, the following four cases may occur:

(a) $0 < c_1, c_2 \leq a_1$ with $c_1 < c_2$,

(b) $0 < c_1 \leq a_1$ and $a_1 < c_2 \leq a_2$,

(c) $0 < c_1, c_2 \leq a_1$ with $c_1 > c_2$,

(d) $0 < c_2 \leq a_1$ and $a_1 < c_1 \leq a_2$.

Note that if the kernel is symmetric with respect to $c_1 < c_2$ and $c_1 > c_2$, only the cases (a) and (b) have to be considered. With (7) and (9) the general form of the kernel matrix $\boldsymbol{P}(c_1, c_2, a_1, a_2)$ can be written as a composition of nine submatrices $\boldsymbol{P}_{ij}(\cdot)$ of appropriate dimension. For the case $c_1 < c_2$ the transition kernel $\boldsymbol{P}(c_1, c_2, a_1, a_2)$ has the form:

$$\boldsymbol{P}(c_1, c_2, a_1, a_2) = \left(\begin{array}{c|c|c} \boldsymbol{P}_{11}(a_1) & \boldsymbol{P}_{12}(a_1, a_2) & \boldsymbol{P}_{13}(a_1, a_2) \\ \hline \boldsymbol{P}_{21}(c_1, a_1) & \boldsymbol{P}_{22}(c_1, a_1, a_2) & \boldsymbol{P}_{23}(c_1, a_1, a_2) \\ \hline \boldsymbol{P}_{31}(c_1, c_2, a_1) & \boldsymbol{P}_{32}(c_1, c_2, a_1, a_2) & \boldsymbol{P}_{33}(c_1, c_2, a_1, a_2) \end{array}\right) \begin{array}{c} 1 \\ \vdots \\ N_1 \\ \hline N_1+1 \\ \vdots \\ N_1+N_2 \\ \hline N_1+N_2+1 \\ \vdots \\ N \end{array} \tag{10}$$

$$1 \quad N_1 \,|\, N_1+1 \quad N_1+N_2 \,|\, N_1+N_2+1 \quad N$$

Taking into account the cases (a) and (b), we distinguish in the submatrices $\boldsymbol{P}_{22}(c_1, a_1, a_2)$, $\boldsymbol{P}_{23}(c_1, a_1, a_2)$, $\boldsymbol{P}_{32}(c_1, c_2, a_1, a_2)$ and $\boldsymbol{P}_{33}(c_1, c_2, a_1, a_2)$ the two cases:

$$\boldsymbol{P}_{22}(c_1, a_1, a_2) = \begin{cases} \boldsymbol{R}_{22}(c_1, a_1, a_2) & \text{for } 0 < c_1 \le a_1, \\ \boldsymbol{S}_{22}(c_1, a_1, a_2) & \text{for } 0 < a_1 \le c_1, \end{cases}$$

$$\boldsymbol{P}_{23}(c_1, a_1, a_2) = \begin{cases} \boldsymbol{R}_{23}(c_1, a_1, a_2) & \text{for } 0 < c_1 \le a_1, \\ \boldsymbol{S}_{23}(c_1, a_1, a_2) & \text{for } 0 < a_1 \le c_1 \le a_2, \end{cases} \tag{11}$$

$$\boldsymbol{P}_{32}(c_1, c_2, a_1, a_2) = \begin{cases} \boldsymbol{R}_{23}(c_1, c_2, a_1, a_2) & \text{for } 0 < c_1, c_2 \le a_1 \text{ with } c_1 < c_2, \\ \boldsymbol{S}_{23}(c_1, c_2, a_1, a_2) & \text{for } 0 < c_1 \le a_1 < c_2, \end{cases}$$

$$\boldsymbol{P}_{33}(c_1, c_2, a_1, a_2) = \begin{cases} \boldsymbol{R}_{33}(c_1, c_2, a_1, a_2) & \text{for } 0 < c_1, c_2 \le a_1 \text{ with } c_1 < c_2, \\ \boldsymbol{S}_{33}(c_1, c_2, a_1, a_2) & \text{for } 0 < c_1 \le a_1 \text{ and } a_1 < c_2 \le a_2. \end{cases} \tag{12}$$

In (10), the submatrix $\boldsymbol{P}_{11}(a_1)$ represents state transitions among states of S_{exp}, $\boldsymbol{P}_{12}(a_1, a_2)$ represents state transitions from states of S_{exp} to states of S_{det1}, and $\boldsymbol{P}_{13}(a_1, a_2)$ represents state transitions from states of S_{exp} to states of S_{det2}. Furthermore, submatrix $\boldsymbol{P}_{22}(c_1, a_1, a_2)$ represents state transitions among states of S_{det1} and $\boldsymbol{P}_{21}(c_1, a_1)$ represents state transitions from states of S_{det1} to states of S_{exp}. The submatrices $\boldsymbol{P}_{23}(c_1, a_1, a_2)$ represents state transitions from states of S_{det1} to states of S_{det2}, respectively. State transition from states of S_{det2} to states of S_{det1} and S_{exp} are represented by the submatrices $\boldsymbol{P}_{32}(c_1, c_2, a_1, a_2)$ and $\boldsymbol{P}_{31}(c_1, c_2, a_1)$. The submatrix $\boldsymbol{P}_{33}(c_1, c_2, a_1, a_2)$ represents state transition among states of S_{det2}. If $c_1 > c_2$ the transition kernel $\boldsymbol{P}(c_1, c_2, a_1, a_2)$ is of similar form. The difference lies in that the submatrices $\boldsymbol{P}_{21}(\cdot)$, $\boldsymbol{P}_{22}(\cdot)$, and $\boldsymbol{P}_{23}(\cdot)$ may depend on c_2 instead of c_1, i.e., for $c_1 > c_2$ these submatrices are of the form $\boldsymbol{P}_{21}(c_2, a_1)$, $\boldsymbol{P}_{22}(c_2, a_1, a_2)$, and $\boldsymbol{P}_{23}(c_2, a_1, a_2)$.

As described in [12,13], the transition kernel of the form (10) can be effectively determined by an extension of the concept of subordinated Markov chains introduced in [11]. Recall that a subordinated Markov chain associated with a state s_i is a CTMC whose states are given by the transitive closure of all states s_k reachable from s_i via the occurrence of exponential events. Numerical computation of kernel elements is performed by transient analysis of these CTMCs [9], though not only for instant of time D (as for steady-state analysis of DSPNs without concurrent deterministic transitions [11]), but also for instants of time $0 < t \leq D$. Thus, numerical computation of the transition kernel $\boldsymbol{P}(c_1, c_2, a_1, a_2)$ requires asymptotically the same effort as the computation of the probability matrix of the embedded DTMC for a DSPN without concurrent deterministic transitions. Furthermore, it is clear that all kernel elements $p_{ij}(\cdot)$ are continuous and differentiable.

A key contribution of this paper constitutes the observation that the transition kernel $\boldsymbol{P}(c_1, c_2, a_1, a_2)$ is *separable*. That means that submatrices of $\boldsymbol{P}(c_1, c_2, a_1, a_2)$ can be expressed as sums of matrices $\boldsymbol{K}_{ij}$ and $\boldsymbol{J}_{ij}$ comprising only constant entries, matrices comprising only functional entries setting new clocks $\boldsymbol{G}_{ij}(\cdot)$ and $\boldsymbol{H}_{ij}(\cdot)$; i.e., comprising only of functional entries in a_1 and/or a_2, and matrices comprising only functional entries taking into account old clocks $\boldsymbol{U}_{ij}(\cdot)$ and $\boldsymbol{V}_{ij}(\cdot)$; i.e., comprising only of functional entries in c_1 and/or c_2. That is:

$$\boldsymbol{P}(c_1, c_2, a_1, a_2) = \left(\begin{array}{c|c|c} \boldsymbol{P}_{11}(a_1) & \boldsymbol{P}_{12}(a_1, a_2) & \boldsymbol{P}_{13}(a_1, a_2) \\ \hline \boldsymbol{K}_{21} + \boldsymbol{U}_{21}(c_1) + \boldsymbol{G}_{21}(a_1) & \boldsymbol{P}_{22}(c_1, a_1, a_2) & \boldsymbol{P}_{23}(c_1, a_1, a_2) \\ \hline \boldsymbol{K}_{31} + \boldsymbol{U}_{31}(c_1, c_2) + \boldsymbol{G}_{31}(a_1) & \boldsymbol{P}_{32}(c_1, c_2, a_1, a_2) & \boldsymbol{P}_{33}(c_1, c_2, a_1, a_2) \end{array}\right) \tag{13}$$

with

$$\boldsymbol{P}_{22}(c_1, a_1, a_2) = \begin{cases} \boldsymbol{K}_{22} + \boldsymbol{U}_{22}(c_1) + \boldsymbol{G}_{22}(a_1, a_2) & \text{for } 0 < c_1 \leq a_1, \\ \boldsymbol{J}_{22} + \boldsymbol{V}_{22}(c_1) + \boldsymbol{H}_{22}(a_1, a_2) & \text{for } 0 < a_1 \leq c_1, \end{cases} \tag{14}$$

$$\boldsymbol{P}_{23}(c_1, a_1, a_2) = \begin{cases} \boldsymbol{K}_{23} + \boldsymbol{U}_{23}(c_1) + \boldsymbol{G}_{23}(a_1, a_2) & \text{for } 0 < c_1 \leq a_1, \\ \boldsymbol{J}_{23} + \boldsymbol{V}_{23}(c_1) + \boldsymbol{H}_{23}(a_1, a_2) & \text{for } 0 < a_1 \leq c_1 \leq a_2. \end{cases} \tag{15}$$

The matrices $\boldsymbol{P}_{32}(c_1, c_2, a_1, a_2)$ and $\boldsymbol{P}_{33}(c_1, c_2, a_1, a_2)$ are separated in a similar way.

The separability of the transition kernel implies that numerical integration in the iterative solution scheme of the Fredholm equations described in Section 4 need only be performed for kernel elements that actually depend on current clock values c_1 and/or c_2. Furthermore, even if numerical integration has to be performed, we can take out from the composite quadrature formulas expressions dependent on new clock values a_1 and/or a_2. Thus, in each iteration for each state probability and mesh point only a constant number of summations are required to perform integration by numerical quadrature.

4. Iterative solution of the Fredholm integral equations

As shown in [12,13] the GSSMC approach allows the numerical analysis of DSPNs with concurrent deterministic transitions with different delays. However, for ease of exposition we discuss in this section only the restricted case that all deterministic transitions of the DSPN have the same firing delay D. The extension of the time-dependent equations for DSPNs with deterministic transitions having different delays can be performed exactly as for the system of stationary equations introduced in [13]. Furthermore, we assume that the transition kernel is symmetric with respect to $c_1 < c_2$ and $c_1 > c_2$. This assumption implies that for $s_j \in S_{\mathrm{det2}}$ $\pi_j^{(n)}(a_1, a_2) = \pi_j^{(n)}(a_2, a_1)$ with $0 < a_1, a_2 \le D$ and $n = 0, 1, 2 \ldots$. Relaxing this assumption replaces the factor 2 in the system (17) to (19) by an additional two-dimensional integral expressions. Furthermore, this system of equations must be extended by an additional equation to compute state probabilities of the form $\pi_j^{(n)}(a_2, a_1)$. These steps are exactly the same as discussed in [12] for the system of stationary equations.

To write the system of time-dependent equations for the GSSMC in vector notation, we define three vectors of state probabilities for the states of S_{exp}, S_{det1}, and, S_{det2} respectively:

$$\begin{aligned}
\pi_{\mathrm{exp}}^{(n)} &= \left(\pi_1^{(n)}, \pi_2^{(n)}, \ldots, \pi_{N_1}^{(n)}\right), \\
\pi_{\mathrm{det1}}^{(n)}(a_1) &= \left(\pi_{N_1+1}^{(n)}(a_1), \pi_{N_1+2}^{(n)}(a_1), \ldots, \pi_{N_1+N_2}^{(n)}(a_1)\right), \\
\pi_{\mathrm{det2}}^{(n)}(a_1, a_2) &= \left(\pi_{N_1+N_2+1}^{(n)}(a_1, a_2), \pi_{N_1+N_2+2}^{(n)}(a_1, a_2), \ldots, \pi_{N}^{(n)}(a_1, a_2)\right).
\end{aligned} \tag{16}$$

Then, using the submatrices $\boldsymbol{P}_{ij}(\cdot)$ of the transition kernel defined in (10) to (12), time-dependent state probabilities for the GSSMC underlying a DSPN with two deterministic transitions concurrently enabled can be derived by the (discrete-time) forward Chapman-Kolmogorov equations. Thus, for $n = 0, 1, 2, \ldots$ we have:

$$\begin{aligned}
\pi_{\mathrm{exp}}^{(n+1)} = \pi_{\mathrm{exp}}^{(n)} \cdot \boldsymbol{P}_{11}(D) &+ \int_0^D \frac{\mathrm{d}\pi_{\mathrm{det1}}^{(n)}(c_1)}{\mathrm{d}c_1} \cdot \boldsymbol{P}_{21}(c_1, D)\, \mathrm{d}c_1 \\
&+ 2 \int_0^D \int_0^{c_2} \frac{\partial^2 \pi_{\mathrm{det2}}^{(n)}(c_1, c_2)}{\partial c_1 \partial c_2} \cdot \boldsymbol{P}_{31}(c_1, c_2, D)\, \mathrm{d}c_1 \mathrm{d}c_2,
\end{aligned} \tag{17}$$

$$\begin{aligned}
\pi_{\mathrm{det1}}^{(n+1)}(a_1) = \pi_{\mathrm{exp}}^{(n)} \cdot \boldsymbol{P}_{12}(a_1, D) &+ \int_0^{a_1} \frac{\mathrm{d}\pi_{\mathrm{det1}}^{(n)}(c_1)}{\mathrm{d}c_1} \cdot \boldsymbol{R}_{22}(c_1, a_1, D)\, \mathrm{d}c_1 \\
&+ \int_{a_1}^D \frac{\mathrm{d}\pi_{\mathrm{det1}}^{(n)}(c_1)}{\mathrm{d}c_1} \cdot \boldsymbol{S}_{22}(c_1, a_1, D)\, \mathrm{d}c_1 \\
&+ 2 \int_0^{a_1} \int_0^{c_2} \frac{\partial^2 \pi_{\mathrm{det2}}^{(n)}(c_1, c_2)}{\partial c_1 \partial c_2} \cdot \boldsymbol{R}_{32}(c_1, c_2, a_1, D)\, \mathrm{d}c_1 \mathrm{d}c_2 \\
&+ 2 \int_{a_1}^D \int_0^{a_1} \frac{\partial^2 \pi_{\mathrm{det2}}^{(n)}(c_1, c_2)}{\partial c_1 \partial c_2} \cdot \boldsymbol{S}_{32}(c_1, c_2, a_1, D)\, \mathrm{d}c_1 \mathrm{d}c_2,
\end{aligned} \tag{18}$$

$$\pi_{\mathrm{det2}}^{(n+1)}(a_1, a_2) = \pi_{\mathrm{exp}}^{(n)} \cdot \boldsymbol{P}_{13}(a_1, a_2) + \int_0^{a_1} \frac{\mathrm{d}\pi_{\mathrm{det1}}^{(n)}(c_1)}{\mathrm{d}c_1} \cdot \boldsymbol{R}_{23}(c_1, a_1, a_2)\,\mathrm{d}c_1$$
$$+ \int_{a_1}^{a_2} \frac{\mathrm{d}\pi_{\mathrm{det1}}^{(n)}(c_1)}{\mathrm{d}c_1} \cdot \boldsymbol{S}_{23}(c_1, a_1, a_2)\,\mathrm{d}c_1$$
$$+ 2\int_0^{a_1}\int_0^{c_2} \frac{\partial^2\pi_{\mathrm{det2}}^{(n)}(c_1, c_2)}{\partial c_1 \partial c_2} \cdot \boldsymbol{R}_{33}(c_1, c_2, a_1, a_2)\,\mathrm{d}c_1\mathrm{d}c_2$$
$$+ 2\int_{a_1}^{a_2}\int_0^{a_1} \frac{\partial^2\pi_{\mathrm{det2}}^{(n)}(c_1, c_2)}{\partial c_1 \partial c_2} \cdot \boldsymbol{S}_{33}(c_1, c_2, a_1, a_2)\,\mathrm{d}c_1\mathrm{d}c_2 \qquad \text{for } 0 < a_1, a_2 \le D. \tag{19}$$

The system of Eqs. (17)–(19) constitutes a system of two-dimensional Fredholm integral equations already written in an iterative scheme for its numerical solution. This iterative scheme called *Picard iteration* is known to converge to a unique solution, if the transition kernel of the GSSMC is continuous or piece-wise continuous in the rectangular region $[0, D] \times [0, D]$ and, thus, $\boldsymbol{P}(c_1, c_2, a_1, a_2)$ is bounded [3]. Due to the decomposition in four different subregions, all elements of $\boldsymbol{P}(c_1, c_2, a_1, a_2)$ as defined in Section 3 are piece-wise continuous. Thus, the iterative scheme (17) to (19) converges to the stationary or time-averaged solution defined in Eq. (6) when n goes to infinity. Moreover, by taking the limits $n \to \infty$ in (17) to (19), we derive the system of stationary equations for the GSSMC underlying a DSPN with two deterministic transitions concurrently enabled introduced in [12,13].

In each iteration $n = 0, 1, 2, \ldots$, the boundary conditions of the system (17) to (19) are given by:

$$\begin{aligned} &\pi_i^{(n)}(0) = 0 && \text{for } N_1 + 1 \le i \le N_1 + N_2, \\ &\pi_i^{(n)}(c_1, 0) = 0 && \text{for } N_1 + N_2 + 1 \le i \le N \text{ and } 0 \le c_1 \le D, \\ &\pi_i^{(n)}(0, c_2) = 0 && \text{for } N_1 + N_2 + 1 \le i \le N \text{ and } 0 \le c_2 \le D. \end{aligned} \tag{20}$$

Note that the system (17) to (19) already includes the normalization condition (21) for the state probabilities of the marking process. For $n = 0, 1, 2, \ldots$ holds:

$$\sum_{j=1}^{N_1} \pi_j^{(n)} + \sum_{j=N_1+1}^{N_1+N_2} \pi_j^{(n)}(D) + \sum_{j=N_1+N_2+1}^{N} \pi_j^{(n)}(D, D) = 1. \tag{21}$$

Using (5), the time-dependent state probabilities at arbitrary instants of time $t = nD + a$ are given by:

$$\begin{aligned} \pi_{\mathrm{exp}}^{(nD+a)} &= \pi_{\mathrm{exp}}^{(n)} \cdot \boldsymbol{P}_{11}(a) + \int_0^{a} \frac{\mathrm{d}\pi_{\mathrm{det1}}^{(n)}(c_1)}{\mathrm{d}c_1} \cdot \boldsymbol{P}_{21}(c_1, a)\,\mathrm{d}c_1 \\ &\quad + 2\int_0^{a}\int_0^{c_2} \frac{\partial^2\pi_{\mathrm{det2}}^{(n)}(c_1, c_2)}{\partial c_1 \partial c_2} \cdot \boldsymbol{P}_{31}(c_1, c_2, a)\,\mathrm{d}c_1\mathrm{d}c_2, \\ \pi_{\mathrm{det1}}^{(nD+a)}(D) &= \pi_{\mathrm{det1}}^{(n+1)}(a), \\ \pi_{\mathrm{det2}}^{(nD+a)}(D, D) &= \pi_{\mathrm{det1}}^{(n+1)}(a, a). \qquad \text{for } 0 < a < D. \end{aligned} \tag{22}$$

Let us point out that if the DSPN contains only exponential transitions (i.e., $S_{\mathrm{det1}} = S_{\mathrm{det2}} = \{\}$) the system (17) to (19) reduces to an ordinary Chapman–Kolmogorov equation in discrete time. That is:

$$\pi_{\exp}^{(n+1)} = \pi_{\exp}^{(n)} \cdot \boldsymbol{P}_{11}(D) \quad \text{where } \boldsymbol{P}_{11}(D) = \mathrm{e}^{QD}. \tag{23}$$

In (23) the matrix $\boldsymbol{Q}$ constitutes the generator of the CTMC defined by tangible markings of such a DSPN (i.e., a GSPN) and state transitions corresponding to firings of exponential transitions.

To simplify the notation in the system (17) to (19), we introduce two vectors $y^{(n)}(c_1)$ and $z^{(n)}(c_1, c_2)$ for the derivatives of state probabilities as:

$$y^{(n)}(c_1) = \frac{\mathrm{d}\pi_{\mathrm{det1}}^{(n)}(c_1)}{\mathrm{d}c_1}, \tag{24}$$

$$z^{(n)}(c_1, c_2) = \frac{\partial^2 \pi_{\mathrm{det2}}^{(n)}(c_1, c_2)}{\partial c_1 \partial c_2}. \tag{25}$$

Using (24) and (25) and exploiting the separability of the transition kernel introduced in (13) to (15), we can rewrite the system (17) to (19) as:

$$\begin{aligned}\pi_{\exp}^{(n+1)} = {} & \pi_{\exp}^{(n)} \cdot \boldsymbol{P}_{11}(D) + \pi_{\mathrm{det1}}^{(n)}(D) \cdot (\boldsymbol{K}_{21} + \boldsymbol{G}_{21}(D)) \\ & + \int_0^D y^{(n)}(c_1) \cdot \boldsymbol{U}_{21}(c_1)\,\mathrm{d}c_1 + \pi_{\mathrm{det2}}^{(n)}(D, D) \cdot (\boldsymbol{K}_{31} + \boldsymbol{G}_{31}(D)) \\ & + 2 \int_0^D \int_0^{c_2} z^{(n)}(c_1, c_2) \cdot \boldsymbol{U}_{31}(c_1, c_2)\,\mathrm{d}c_1 \mathrm{d}c_2,\end{aligned} \tag{26}$$

$$\begin{aligned}\pi_{\mathrm{det1}}^{(n+1)}(a_1) = {} & \pi_{\exp}^{(n)} \cdot \boldsymbol{P}_{12}(a_1) + \pi_{\mathrm{det1}}^{(n)}(a_1) \cdot (\boldsymbol{K}_{22} + \boldsymbol{G}_{22}(a_1, D)) \\ & + \int_0^{a_1} y^{(n)}(c_1) \cdot \boldsymbol{U}_{22}(c_1)\,\mathrm{d}c_1 + \left(\pi_{\mathrm{det1}}^{(n)}(D) - \pi_{\mathrm{det1}}^{(n)}(a_1)\right) \cdot (\boldsymbol{J}_{22} + \boldsymbol{H}_{22}(a_1, D)) \\ & + \int_{a_1}^D y^{(n)}(c_1) \cdot \boldsymbol{V}_{22}(c_1)\,\mathrm{d}c_1 + \pi_{\mathrm{det2}}^{(n)}(a_1, a_1) \cdot (\boldsymbol{K}_{32} + \boldsymbol{G}_{32}(a_1, D)) \\ & + 2 \int_0^{a_1} \int_0^{c_2} z^{(n)}(c_1, c_2) \cdot \boldsymbol{U}_{32}(c_1, c_2)\,\mathrm{d}c_1 \mathrm{d}c_2 + 2 \left(\pi_{\mathrm{det2}}^{(n)}(a_1, D) - \pi_{\mathrm{det2}}^{(n)}(a_1, a_1)\right) \\ & \quad \cdot (\boldsymbol{J}_{32} + \boldsymbol{H}_{32}(a_1, D)) \\ & + 2 \int_{a_1}^D \int_0^{a_1} z^{(n)}(c_1, c_2) \cdot \boldsymbol{V}_{32}(c_1, c_2)\,\mathrm{d}c_1 \mathrm{d}c_2,\end{aligned} \tag{27}$$

$$
\begin{aligned}
\pi_{\mathrm{det2}}^{(n+1)}(a_1,a_2) &= \pi_{\exp}^{(n)} \cdot \boldsymbol{P}_{13}(a_1,a_2) + \pi_{\mathrm{det1}}^{(n)}(a_1)\cdot(\boldsymbol{K}_{23}+\boldsymbol{G}_{23}(a_1,a_2)) \\
&\quad + \int_0^{a_1} y^{(n)}(c_1)\cdot \boldsymbol{U}_{23}(c_1)\,\mathrm{d}c_1 + \left(\pi_{\mathrm{det1}}^{(n)}(a_2) - \pi_{\mathrm{det1}}^{(n)}(a_1)\right)\cdot(\boldsymbol{J}_{23}+\boldsymbol{H}_{23}(a_1,a_2)) \\
&\quad + \int_{a_1}^{a_2} y^{(n)}(c_1)\cdot \boldsymbol{V}_{23}(c_1)\,\mathrm{d}c_1 + \pi_{\mathrm{det2}}^{(n)}(a_1,a_1)\cdot(\boldsymbol{K}_{33}+\boldsymbol{G}_{33}(a_1,a_2)) \\
&\quad + 2\int_0^{a_1}\int_0^{c_2} z^{(n)}(c_1,c_2)\cdot \boldsymbol{U}_{33}(c_1,c_2)\,\mathrm{d}c_1\mathrm{d}c_2 \\
&\quad + 2\left(\pi_{\mathrm{det2}}^{(n)}(a_1,a_2) - \pi_{\mathrm{det2}}^{(n)}(a_1,a_1)\right)\cdot(\boldsymbol{J}_{33}+\boldsymbol{H}_{33}(a_1,a_2)) \\
&\quad + 2\int_{a_1}^{a_2}\int_0^{a_1} z^{(n)}(c_1,c_2)\cdot \boldsymbol{V}_{33}(c_1,c_2)\,\mathrm{d}c_1\mathrm{d}c_2 \qquad \text{for } 0 < a_1, a_2 \le D. \qquad (28)
\end{aligned}
$$

The system of Fredholm Eqs. (26)–(28) together with (24) and (25) give rise to an iterative scheme for the effective numerical transient analysis of the marking process of a DSPN in which two deterministic transitions may be enabled concurrently.

Using an appropriate discretization, the iterative scheme (24) to (28) leads to a constant number of additions and a vector-matrix multiplication for each mesh point $(k\Delta, l\Delta)$ with $1 \le k, l \le M$. The number of discretization steps (in each direction), denoted by M, is given by $M = D/\Delta$. Depending on the (mean) firing delays of timed transitions of a DSPN, the value of Δ can be automatically determined such that a pre-defined error tolerance for the approximation of integrals is met [3]. Using an appropriate formula for numerical differentiation, in each iteration for each mesh point the vector $y^{(n)}(k\Delta)$ and $z^{(n)}(k\Delta, l\Delta)$ of Eq. (24) is computed with a constant number of additions.

Subsequently, each integral in the system of Fredholm Eqs. (26)–(28) is approximated by a finite sum that corresponds to the integral of an interpolating polynomial over some partition of the interval of integration. In our current implementation, we employ a composite Simpson rule combined with a 3/8

Table 1
Main steps of the iterative scheme for numerical transient analysis of DSPNs

for $n = 0, 1, 2, \ldots, n_0 - 1$
- (1) for $k = 0, 1, \ldots, M - 1$
  - compute the vector of derivatives $y^{(n)}(k\Delta)$ according to (24)
- (2) for $k = 0, 1, \ldots, M$ and $l = k, k + 1, \ldots, M$
  - compute the vector of derivatives $z^{(n)}(k\Delta, l\Delta)$ according to (25)
- (3) perform next iteration step
  - (3.1) compute the vector of state probabilities $\pi_{\exp}^{(n+1)}$ according to (26)
  - (3.2) for $k = 1, 2, \ldots, M$
    - compute the vector of state probabilities $\pi_{\mathrm{det1}}^{(n+1)}(k\Delta)$ according to (27)
  - (3.3) for $k = 1, 2, \ldots, M$ and $l = k, k + 1, \ldots, M$
    - compute the vector of state probabilities $\pi_{\mathrm{det2}}^{(n+1)}(k\Delta, l\Delta)$ according to (28)

rule for both one-dimensional and two-dimensional integrals as a straightforward quadrature formula with fixed step size [12].

Given the submatrices of the transition kernel $\boldsymbol{P}(c_1, c_2, a_1, a_2)$ of (13) to (15) and the initial distribution of the GSSMC according to (3) at each mesh point $(k\Delta, l\Delta)$ with $1 \le k, l \le M$ and the mission time $t = n_0 D$, the main steps of the iterative scheme are summarized in Table 1.

5. Performance curves

To illustrate the practical applicability of the GSSMC approach for transient analysis of DSPNs, we consider DSPNs of two queueing systems of high interest for communication network performance analysis. For these two DSPNs we present curves for CPU solution time and memory requirements versus model size. The experiments have been performed on a Sun Sparc Enterprise station with 1 GByte main memory running the operating system SunOS5.6. For the performance tests the CPU time has been measured by the UNIX system call *clock*.

Fig. 1 shows a DSPN of an MMPP/D/2/K queue as an example for a DSPN with concurrent

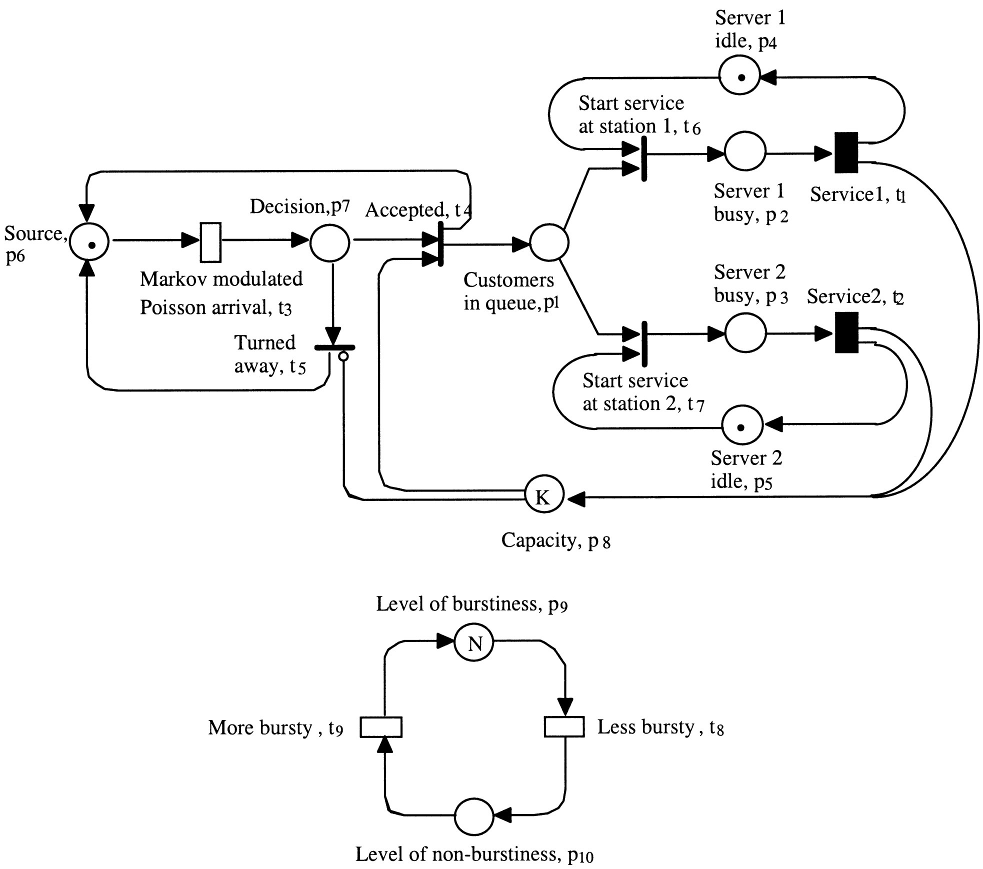


Fig. 1. DSPN of the MMPP/D/2/K queue.

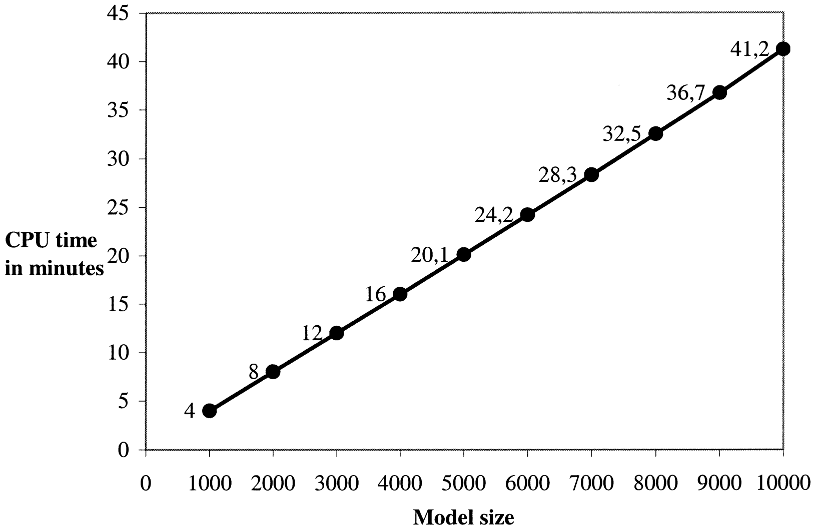


Fig. 2. MMPP/D/2/K queue: CPU time vs. model size.

deterministic transitions. The K tokens residing in place *Capacity* in the initial marking represent the finite number of buffers of the queueing system. The number tokens residing the place *Level of burstiness* control the mean firing time of the exponential transition *Markov modulated Poisson arrival.* That is, the Markov modulated Poisson arrival stream is represented by defining the firing delay of the corresponding exponential transition dependent on the number of tokens in the place *Level of burstiness*. Tokens contained in the places *Customers in queue* represent customers waiting in the queue. Tokens contained in the places *Server 1 busy* and *Server 2 busy* represent customers currently being served. The number of tangible markings of this DSPN is given by $(K + 2) \cdot (N + 1)$. The constant service requirements are modeled by the deterministic transitions *Service 1* and *Service 2* which have firing delay $D = 1.0$. We assume that the immediate transitions *Start service at station 1* and *Start service at station 2* have both associated firing weights 1/2, such that arriving customers to an empty system join each server with equal probability.

In all experiments, model parameters of the arrival process are set such that the effective arrival rate $\lambda_{\mathrm{eff}} = 0.9$ and zero customers reside in the queue at time $t = 0$. The number of discretization steps employed in each dimension in the composite quadrature rule in the iterative scheme is $M = 10$.

Fig. 2 plots the CPU time required for computing the transient solution at instant of time $t = 100$ for increasing model size. We observe a linear growth of CPU time. This is due to the exploitation of the separabilty of the transition kernel $\boldsymbol{P}(c_1, c_2, a_1, a_2)$ resulting in an almost linear growth of the nonzero kernel elements to be considered in the iterative scheme. Fig. 3 plots the memory requirements for storing the nonzero elements of the transition kernel versus model size and, thus, provides further evidence along this line. In a second experiment, the model size is kept fixed to 5000 and the mission time (i.e., the number of iterations that have to be performed by the iterative scheme) is varied. As expected, Fig. 4 shows a linear growth of CPU time for increasing mission time, since in each step of the iterative scheme a constant number of vector matrix multiplications is performed.

Fig. 5 shows a DSPN for an MMPP/D/1/K queue with breakdown and repair as an example for a DSPN without concurrent deterministic transitions. The difference to the DSPN of Fig. 1 lies in that the

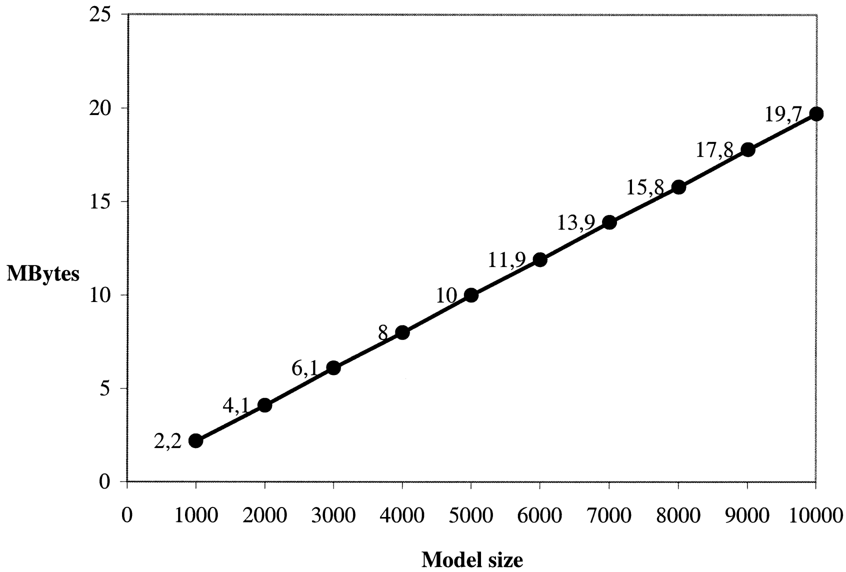


Fig. 3. MMPP/D/2/K queue: memory requirements vs. model size.

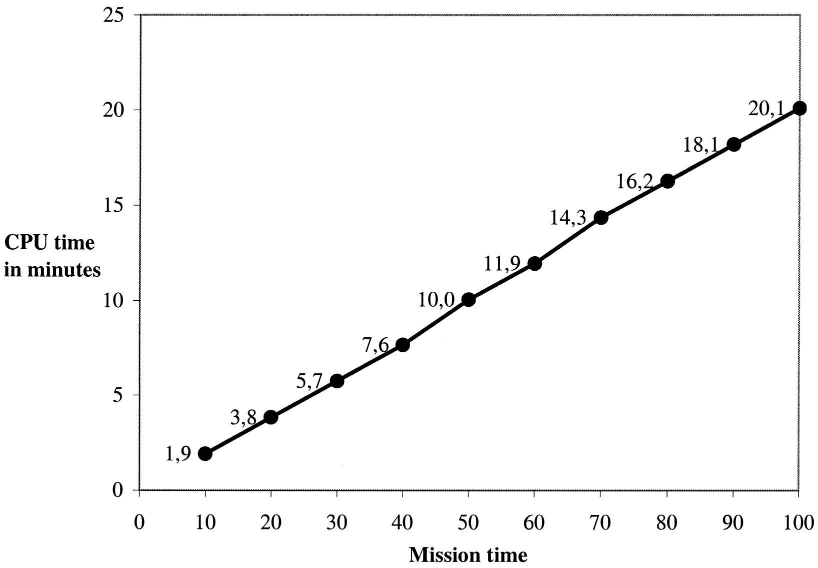


Fig. 4. MMPP/D/2/K queue: CPU time vs. mission time.

system contains only one service center, which may fail and can be repaired. The number of tangible markings of this DSPN is given by $2 \cdot (K+1) \cdot (N+1)$. As in Fig. 1, the constant service requirement is assumed as $D = 1.0$. In all experiments, model parameters of the Markov modulated arrival process are set such that the effective arrival rate $\lambda_{\text{eff}} = 0.9$, zero customers reside in the queue and the system is operating at time $t = 0$. Failures of the system are assumed to be exponentially distributed. Repair times are assumed to be constant.

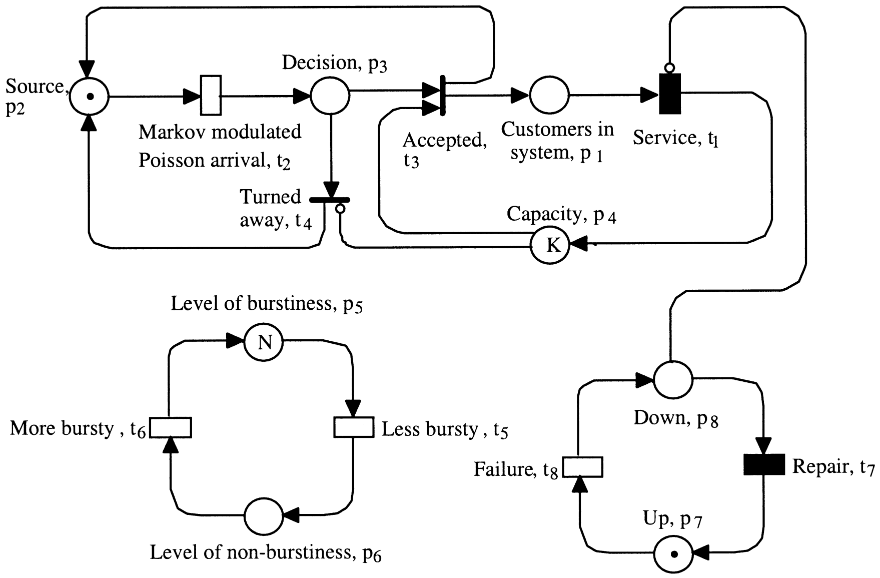


Fig. 5. DSPN of the MMPP/D/1/K queue with breakdown and repair.

Figs. 6–8 show the same set of curves for this DSPN as Figs. 2–4 for the MMPP/D/2/K queue. Again, the number of discretization steps employed in the iterative scheme is $M = 10$. In Figs. 6 and 7, a mission time of $t = 100$ is considered and in Fig. 8 the model size is fixed to 5000. Note that the curves shown in these figures have the same shape as corresponding curves shown in Figs. 2–4. In particular, from Fig. 6 plotting the CPU time versus model size, we observe only a linear growth of

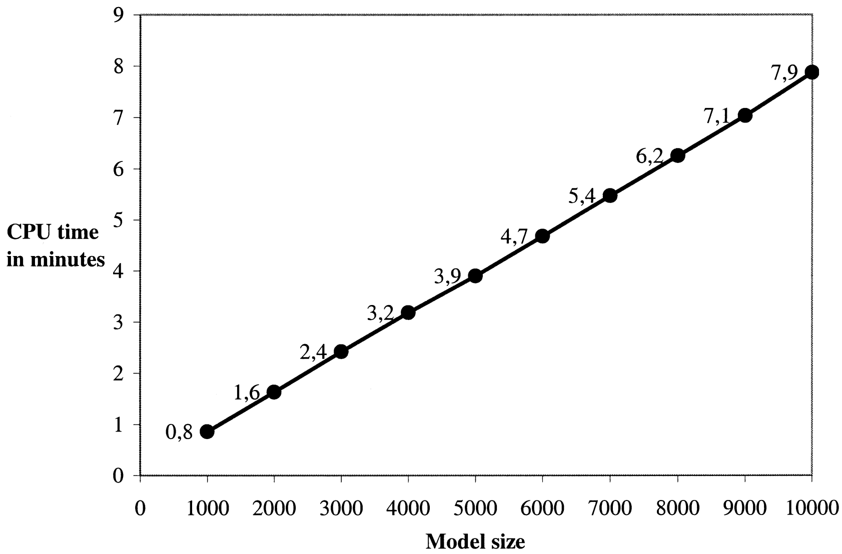


Fig. 6. MMPP/D/1/K queue: CPU time vs. model size.

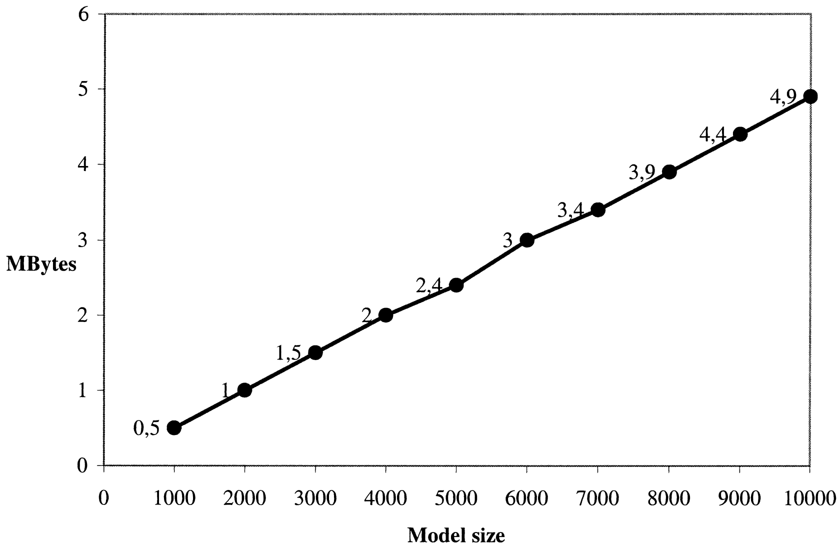


Fig. 7. MMPP/D/1/K queue: memory requirements vs. model size.

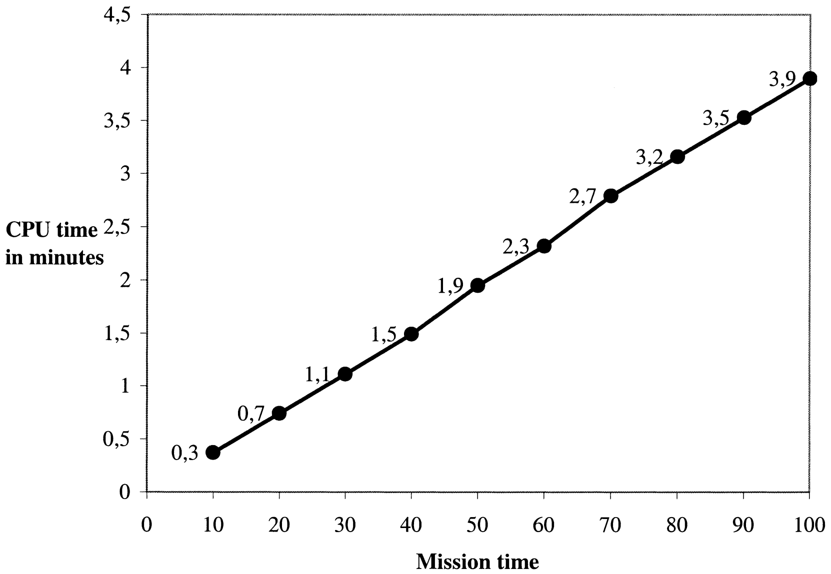


Fig. 8. MMPP/D/1/K queue: CPU time vs. mission time.

CPU time for increasing model size. This again illustrate the benefit of exploiting the separability of the transition kernel in the iterative scheme. A DSPN of an MMPP/D/1/K queue with failure and repair was already considered in [10]. Figs. 6–8 evidently illustrate the computational benefits of the GSSMC approach versus the approach based on the method of supplementary variables proposed in [6,10].

Since the DSPN of Fig. 5 does not contain concurrently enabled deterministic transitions, the stationary of time-averaged state probabilities of its marking process can be computed by an embedded

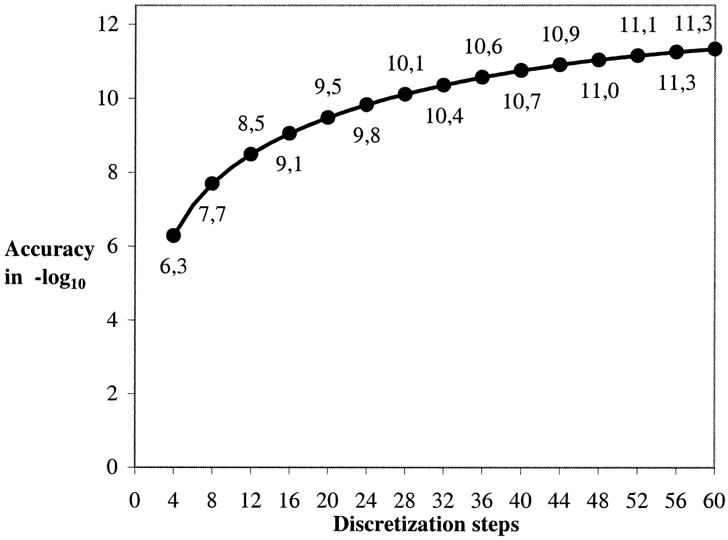


Fig. 9. MMPP/D/1/K queue: accuracy vs. discretization steps.

Markov chain as proposed in [2] and implemented in the software package DSPNexpress [12,14]. We use this fact for estimating the numerical accuracy achieved by the GSSMC approach for a given numerical quadrature of the integral expressions. Fig. 9 plots the accuracy of the stationary distribution of the DSPN of Fig. 5 achieved by the GSSMC approach versus the number of discretization points employed in the iterative scheme. We observe that for already 10 discretization points an accuracy of less that 10^{-7} is obtained.

6. Conclusions

This paper introduced the GSSMC approach for transient analysis of DSPNs with concurrently enabled deterministic transitions. The GSSMC approach is based on numerical iterative solution of a system of Fredholm integral equations. A key contribution of this paper is the observation that the transition kernel of the GSSMC is separable. That is the functional matrix $\boldsymbol{P}(c_1, c_2, a_1, a_2)$ can be expressed as the sum of matrices comprising only constant entries, matrices comprising only of functional entries in a_1 and/or a_2, and matrices comprising only functional entries in c_1 and/or c_2.

To illustrate the practical applicability of the GSSMC approach for transient analysis of large DSPNs with concurrent deterministic transitions, we presented curves for an MMPP/D/2/K queue plotting the CPU time and memory requirements versus model size and mission time, respectively. In order to compare the performance of the GSSMC approach with the supplementary variables approach introduced in previous work [5,10], we provided the same set of curves for an MMPP/D/1/K queue with failure and repair. For this DSPN, the GSSMC approach requires a couple of minutes of CPU time whereas as reported in [10] the supplementary variables approach requires more than 100 hours of CPU time. Thus, we conclude that the GSSMC approach performs numerical transient analysis of large DSPNs without concurrent deterministic transitions three orders of magnitude faster than the previously known approach based on the method of supplementary variables.

References

[1] M. Ajmone Marsan, G. Balbo, A. Bobbio, G. Chiola, G. Conte, A. Cumani, The effect of execution policies on the semantics of stochastic Petri nets, IEEE Trans. Softw. Eng. 15 (1989) 832–845.

[2] M. Ajmone Marsan, G. Chiola, On Petri nets with deterministic and exponentially distributed firing times, in: G. Rozenberg (Ed.), Advances in Petri Nets 1987, Lecture Notes in Computer Science 266, Springer, Berlin, 1987, pp. 132–145.

[3] H. Brunner, P.J. van der Houwen, Numerical Solution of Volterra Equations, North-Holland, Amsterdam, 1986.

[4] H. Choi, V.G. Kulkarni, K.S. Trivedi, Transient analysis of deterministic and stochastic Petri nets, in: M. Ajmone Marsan (Ed.), Application and Theory of Petri Nets 1993, Lecture Notes in Computer Science 691, Springer, Berlin, 1993, pp. 166–185.

[5] R. German, Transient analysis of deterministic and stochastic Petri nets by the method of supplementary variables, in: Proc. 3rd Int. Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, Durham North Carolina, 1995, pp. 394–398.

[6] R. German, C. Lindemann, Analysis of stochastic Petri nets by the method of supplementary variables, in: Proc. Performance '93, Performance Evaluation 20, Special Issue, (1994) 317–335.

[7] P. Glasserman, D.D. Yao, Monotonicity in generalized semi-Markov processes, Math. Oper. Res. 17 (1992) 1–21.

[8] P.W. Glynn, A GSMP formalism for discrete-event systems, Proc. IEEE 77 (1989) 14–23.

[9] D. Gross, D.R. Miller, The randomization technique as a modeling tool and solution procedure for transient Markov processes, Oper. Res. 32 (1984) 345–361.

[10] A. Heindl, R. German, A fourth order algorithm with automatic step size control for the transient analysis of DSPNs, in: Proc. 7th Int. Workshop on Petri Nets and Performance Models, Saint Malo, France, 1997, pp. 60–69.

[11] Ch. Lindemann, An improved numerical algorithm for calculating steady-state solutions of deterministic and stochastic petri net models, Performance Evaluation 18 (1993) 75–91.

[12] Ch. Lindemann, Performance Modelling with Deterministic and Stochastic Petri Nets, Wiley, New York, 1998.

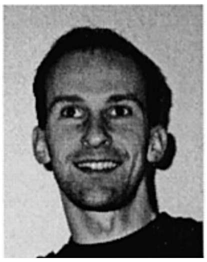
[13] Ch. Lindemann, G.S. Shedler, Numerical analysis of deterministic and stochastic Petri nets with concurrent deterministic transitions, in: Proc. Performance '96, Performance Evaluation 27/28, Special Issue, (1996) 565–582.

[14] Ch. Lindemann, A. Reuys, A. Thümmler, The DSPNexpress 2.000 performance and dependability modeling environment, in: Proc. 29th Int. Symp. on Fault-Tolerant Computing, Madison, WI, 1999 (in press).

[15] D. Logothetis, K.S. Trivedi, Time-dependent behavior of redundant systems with deterministic repair, in: W.J. Stewart (Ed.) Computations with Markov Chains, Kluwer, Dordrecht, 1995, pp. 135–150.

[16] M. Telek, A. Horvath, Transient analysis of age MRSPNs by the method of supplementary variables, in: Proc. 4th Int. Symp. on Parallel and Distributed Systems, Durham, North Carolina, 1998.

Christoph Lindemann is an Associate Professor in the Department of Computer Science at the University of Dortmund and leading the Computer Systems and Performance Evaluation group. From 1994 to 1997, he was a Senior Research Scientist at the GMD Institute for Computer Architecture and Software Technology (GMD FIRST) in Berlin. In the summer 1993 and during the academic year 1994/1995, he was a Visiting Scientist at the IBM Almaden Research Center, San Jose, CA. Christoph Lindemann is a Senior Member of the IEEE. He served on the program committees of various well-known international conferences and he was the program committee co-chair of the 6th International Workshop on Petri Nets and Performance Models, 1995. His research interests include performance and dependability modeling, quantitative system design, parallel and distributed systems, and applied probability.



Axel Thümmler received the degree Diplom-Informatiker (M.S. in Computer Science) from the University of Dortmund in 1998. Presently, he is a Ph.D. student in the Computer Systems and Performance Evaluation group at the University of Dortmund. His research interests include applied probability, algorithm design, and scientific computing.