Numerical Analysis of Stochastic Marked Graph Nets

Peter Buchholz, Peter Kemper
Informatik IV
Universität Dortmund
D-44221 Dortmund, Germany

Abstract

The analysis of stochastic marked graphs is considered. The underlying idea is to decompose the marked graph into subnets, to generate state spaces and transition matrices for these isolated parts and then to represent the generator matrix underlying the complete net by means of much smaller subnet matrices combined via tensor operations. Based on this matrix representation efficient numerical analysis techniques can be used to compute the stationary solution. Furthermore we propose an approximation technique which is similar to known approximate solution techniques for this kind of nets, but our approach is completely integrated in the structured description of the generator matrix. This allows an estimation of the approximation error and the usage of the approximate results as an initial guess for a subsequent iterative analysis, such that the number of required iterations is often significantly reduced.

1 Introduction

The set of stochastic marked graph nets (SMGs) is a well-known subclass of stochastic Petri nets, which allows for concurrency and synchronization but not for decisions. SMGs can be interpreted as a weak extension of fork join queueing networks.

As any other stochastic Petri net the numerical analysis of the continuous time Markov chain (CTMC) underlying a SMG suffers from the state space explosion problem, which excludes applicability of exact conventional analysis methods for large models. Consequently approximate analysis techniques were developed by several authors, see, e.g. [3, 4, 7, 8]. A general characteristic of these techniques is that they are fast and efficient for the price of giving approximate results of unknown accuracy. The latter reduces the confidence in the results significantly, even if techniques have shown to yield sufficiently accurate results for several examples. Approximate analysis techniques can be used for two applications in the context of performance analysis: 1. to derive results where exact techniques are too costly or not applicable at all or 2. to derive a good initial distribution for exact iterative techniques. The published approaches for SMGs have been applied for 1., but, to the best of our knowledge, not for 2.

Due to the state space explosion the main drawback of conventional Markov chain analysis is the size of the stochastic generator matrix **Q**. For certain modelling formalisms it is possible to represent **Q** by tensor sums and products of small matrices (e.g., [1, 2, 6, 11]), which enlarges the size of solvable models by about one order of magnitude. In this paper we describe such a structured matrix representation for SMGs and consequently yield an iterative numerical analysis technique which enlarges the set of solvable models. Solutions of this techniques are exact, as far as iterative numerical solution of CTMCs is exact.

In addition to this result we show how this structured description can be used as a framework to consider approximate analysis techniques; algorithms similar to [3, 7, 8] appear quite naturally in the context of our structured description. This yields the following desired effect.

The quality of an approximate result can be easily measured by the residuals obtained by a single matrix vector calculation which is computable even for large models with several millions of states. If accuracy is not sufficient, results of the approximation can be used as a good initial distribution to proceed with our exact iterative numerical technique which exploits the tensor based representation of the matrix.

The paper is organized as follows. In Sec. 2 basic definitions and an appropriate aggregation are given, which are then used in Sec. 3 to establish a structured description of generator matrix **Q** and to outline the usage of iterative techniques exploiting this structure. Approximate analysis and the relationship of the proposed approach to different approximate analysis techniques known from literature are discussed in Sec. 4. Finally the usefulness and applicability of our approach is demonstrated by an example in Sec. 5.

2 Appropriate aggregations for structured descriptions

The notation follows [3], thus we only briefly introduce basic notations on P/T nets and SMGs, for details we refer to the literature [3, 10]. $\mathcal{N} = (P, T, F)$ is a net if P and T are disjoint sets of places and transitions and $F \subseteq (P \times T) \cup (T \times P)$. We consider nets consisting of non-empty and finite sets of places and

Proceedings of the Sixth International Workshop on Petri Nets and Performance Models (PNPM '95)
1063-6714/95 $10.00 © 1995 IEEE

transitions, respectively. A path of $\mathcal{N} = (P, T, F)$ is a sequence $x_1 \ldots x_k$ with $x_i \in P \cup T$ and $(x_{i-1}, x_i) \in F$ for $1 < i \leq k$. A path is a circuit if $x_1 = x_k$. A path or circuit is simple if all its elements are disjoint (apart from x_1, x_k in case of a circuit). In the sequel we consider only simple paths and circuits. A fundamental circuit is a concatenation of paths such that the resulting sequence is simple and fulfills $x_1 = x_k$ but it need not be a path itself, i.e. some paths inside the sequence might be in opposite direction. The set $\mathcal{P}(x, y)$ denotes the set of all simple paths from x to y. This notation can be extended to sets. Let $X, Y \subseteq P \cup T$, then $\mathcal{P}(X, Y) = \cup_{x \in X} \cup_{y \in Y} \mathcal{P}(x, y)$. A net is strongly connected if for every two elements $x, y \in P \cup T : \mathcal{P}(x, y) \neq \emptyset$. Pre- and post-sets of elements are denoted by $\bullet x = \{y | (y, x) \in F\}$ and $x\bullet = \{y | (x, y) \in F\}$. The notation can also be extended to sets of elements.

A marking is a function $M : P \rightarrow \mathbb{N}$ which is usually represented in vector form. $M[p]$ denotes the number of tokens on place p. A net system is a tuple $(\mathcal{N}, M_0)$ where M_0 is the initial marking. Transition $t \in T$ is enabled in marking M, if $M[p] > 0$ for all $p \in \bullet t$. A transition t enabled in marking M can fire yielding a new marking M' with $M'[p] = M[p] - \delta(p \in \bullet t) + \delta(p \in t\bullet)$, where $\delta(x \in X) = 1$ for $x \in X$ and 0 otherwise. This is denoted as $M[t > M'$, for a sequence $\sigma = t_1 t_2 \ldots t_k$ with $M_1[t_1 > M_2[t_2 > \ldots [t_k > M_{k+1}$ we use the abbreviation $M_1[\sigma > M_{k+1}$, marking M_{k+1} is reachable from M_1 by firing σ. The reachability set $R(\mathcal{N}, M_0)$ is the set of all markings reachable from M_0. We consider in this paper only nets with finite reachability sets, which implies that the possible number of tokens on all places is bounded. A net system is live, if for every reachable marking M and every transition t holds: $\exists \sigma : M[\sigma > M' \wedge M'[t >$. The language $L(\mathcal{N}, M_0)$ of a net system is defined as the set of all firing sequences $\{\sigma | M_0[\sigma >\}$.

A marked graph (MG) is a Petri net, where all places have exactly one input- and one output transition (i.e., $|\bullet p| = |p\bullet| = 1$). We consider here stochastic marked graphs (SMGs), which are MGs, where every transition has an exponentially distributed firing delay with mean value $(\lambda_t)^{-1}$, which might depend, in a restricted way, on the marking of the net. Allowed restrictions are outlined below.

Definition 1 *Let $\mathcal{N} = (P, T, F)$ be a strongly connected MG. A subset of places $Q \subseteq P$, is said to be a cut of $\mathcal{N}$ iff there exist two subnets, $\mathcal{N}_1 = (P_1, T_1, F_1)$ and $\mathcal{N}_2 = (P_2, T_2, F_2)$, of $\mathcal{N}$ verifying*

1. $T_1 \cup T_2 = T$, $T_1 \cap T_2 = \emptyset$
2. $P_1 = \bullet T_1 \cup T_1 \bullet, P_2 = \bullet T_2 \cup T_2 \bullet$
3. $P_1 \cup P_2 = P$, $P_1 \cap P_2 = Q$
4. $F_i = F \cap ((P_i \times T_i) \cup (T_i \times P_i)), i \in \{1, 2\}$

$Q_i^{in} := \bullet T_i \cap Q$ $(Q_i^{out} := T_i \bullet \cap Q)$ *for $i \in \{1, 2\}$ is the set of input (output) places of subnet N_i, $T_i^{in} := \bullet Q_i^{in}$ $(T_i^{out} := \bullet Q_i^{out})$ is the set of input (output) transitions for $\mathcal{N}_i$.*

According to the above definition and the properties of MGs $Q_1^{in} \cap Q_2^{in} = \emptyset$ and $Q_1^{in} \cup Q_2^{in} = Q$. A marking structurally implicit place (MSIP) is a place which is never the unique reason to restrict the firing of its output transitions independently from an initial marking M_0. The main characteristic of a MSIP p is that there is a path from $\bullet p$ to $p\bullet$ circumventing p. Such a path can be described by a vector $Y \in \mathbb{N}_0^P$.

Theorem 1 ([3]) *Let $\mathcal{N} = (P, T, F)$ be a net where $p \notin P$ is added to $\mathcal{N}$ yielding a net $\mathcal{N}^p$ with incidence vector l_p for p. Place p is an MSIP in $\mathcal{N}^p$ iff there exists $Y \geq 0$ such that $Y^T \mathbf{C} = l_p$, where $\mathbf{C}$ is the incidence matrix of $\mathcal{N}$.*

We are interested in adding MSIPs onto a MG such that the resulting net is a MG again. These MSIPs are called TT-MSIPs, since they have exactly one input and one output transition.

Theorem 2 ([3]) *Let $\mathcal{N} = (P, T, F)$ be a strongly connected MG and $p \notin P$ be a place added with one input transition $t_i \in T(\bullet p = \{t_i\})$ and one output transition $t_o \in T(p\bullet = \{t_o\})$. Place p is a TT-MSIP with respect to $\mathcal{N}$ and $\forall \pi \in \mathcal{P}(t_i, t_o) : l_p = \sum_{p_j \in \pi} l_{p_j}$. p is an implicit place in $(\mathcal{N}^p, M_0^p)$ for all initial markings which satisfy $M_0[p] \geq M_0^{min}[p]$. This minimal initial number of tokens on p is defined by $M_0^{min}[p] := min\{\sum_{p_j \in \pi} M_0[p_j] | \pi \in \mathcal{P}(t_i, t_o)\}$.*

The following definition of aggregable parts differs from the aggregable subnets considered in [3] in that it integrates Q_i^{in} into the subnet which will be aggregated. For these reasons we use a different terminology than [3] for the different parts.

Definition 2 *Let $(\mathcal{N}, M_0)$ be a strongly connected, live MG, $Q \subseteq P$ be a cut of N.*

The subnets $\mathcal{N}_{A_i} = (P_{A_i}, T_{A_i}, F_{A_i})$ for $i \in \{1, 2\}$ are called the aggregable parts of cut Q, where

1. $P_{A_i} = P_i \backslash Q_i^{out}$
2. $T_{A_i} = T_i \backslash T_i^{out}$
3. $F_{A_i} = F_i \cap ((P_{A_i} \times T_{A_i}) \cup (T_{A_i} \times P_{A_i}))$

Places in $p \in P_{A_i}$ with $\bullet p \cap T_{A_i} = \emptyset$ are source places and places $p \in P_{A_i}$ with $p\bullet \cap T_{A_i} = \emptyset$ are sink places of N_{A_i}. $\mathcal{P}_{A_i}$ denotes the set of paths in $\mathcal{N}_{A_i}$ from a source place to a sink place. IP_{A_i} is the set of TT-MSIPs which result from $\mathcal{P}_{A_i}$. For any $\pi \in \mathcal{P}_{A_i}$ we introduce a TT-MSIP $p_\pi \in IP_{A_i}$ given by the linear combinations of the rows in the incidence matrix corresponding to the places in π. We denote

Proceedings of the Sixth International Workshop on Petri Nets and Performance Models (PNPM '95)
1063-6714/95 $10.00 © 1995 IEEE

$IP = IP_{A_1} \cup IP_{A_2}$. *The initial marking is denoted as* M_0^a *and is given by* $M_0^a[p] := min\{\sum_{p_j \in \pi} M_0[p_j] | l_p = \sum_{p_j \in \pi} l_{p_j} \wedge \pi \in \mathcal{P}_{A_i} \ \}$ *if* $p \in IP$ *and* $M_0^a[p] := M_0[p]$ *otherwise.*

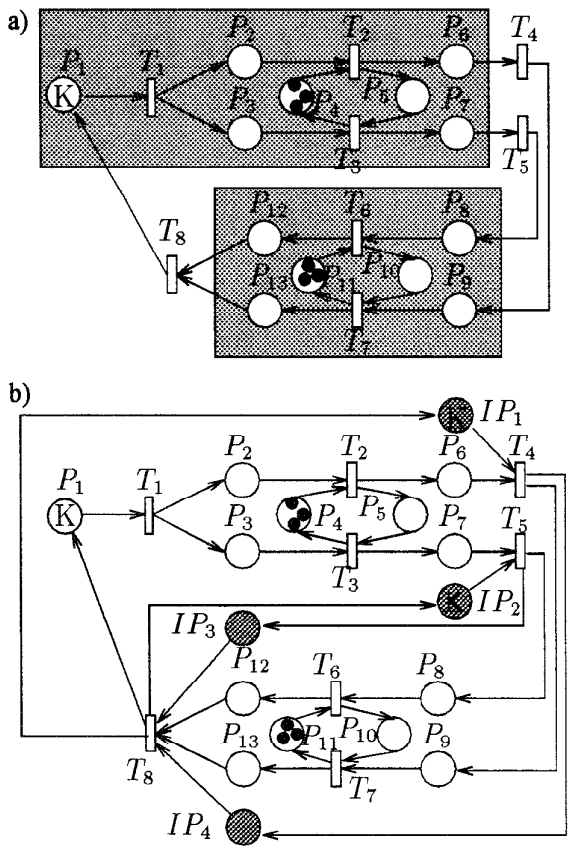

Figure 1: a) example SMG and b) extended system

Observe that $\mathcal{N}_{A_i} = (P_{A_i}, T_{A_i}, F_{A_i})$ does not contain output transitions T_i^{out} and output places included in the cut. Figure 1 a) shows an example net with a cut $Q = \{P_1, P_8, P_9\}$ where each resulting aggregable part is contained in a shaded box.

According to Theorem 2 TT-MSIPs with a sufficient initial marking can be added to a net system without affecting its reachability set or language. The interesting point is that even substitution of an aggregable part by its set IP_{A_i} still yields a reachability set and a language which are equivalent up to the missing transitions, resp. places. Such a substitution can be seen as an abstraction.

Definition 3 *Let* $(\mathcal{N}, M_0)$ *be a strongly connected, live MG,* $Q \subseteq P$ *be a cut of N yielding two subnets* $\mathcal{N}_1 = (P_1, T_1, F_1)$ *and* $\mathcal{N}_2 = (P_2, T_2, F_2)$.

Extended system $ES = (EN, M_0^{EN})$: *ES is derived by adding sets* IP_{A_i}, *for* $i \in \{1,2\}$ *to N and defining* M_0^{EN} *according to Def. (2).*

Low level system $LS_i = (LN_i, M_0^{LN_i})$, $i \in \{1,2\}$: LS_i *is derived by subtracting* $\mathcal{N}_{A_j} = (P_{A_j}, T_{A_j}, F_{A_j})$, $j \neq i$ *from ES, i.e.* $P_{LN_i} := P_{EN} \backslash P_{A_j}$, $T_{LN_i} := T_{EN} \backslash T_{N_{A_j}}$, $F_{LN_i} := F_{EN} \cap ((P_{LN_i} \times T_{LN_i}) \cup (T_{LN_i} \times P_{LN_i}))$, *and* $M_0^{LS_i} := M_0^{ES}|_{P_{LN_i}}$.

High level system $HS = (HN, M_0^{HN})$: *HS derived by subtracting* N_{A_i} *for an* $i \in \{1,2\}$ *from* LS_i *in the same way as* LS_i *was derived from ES.*

In other words the high level system remains after subtracting both aggregable parts from the extended system. Note that HS and $\mathcal{N}$ have only $\bullet Q$ in common, but do not share any places[1]. MSIPs are generated for all shortest paths from input to output places in a subnet. Figure 1 shows an example SMG and the extended system for a cut $Q = \{P_1, P_8, P_9\}$, where the additional MSIPs are shaded. The corresponding low level systems LN_1, LN_2 and the high level system are given in Fig. 2 in Sec. 5.

Theorem 3 *Let* $(\mathcal{N}, M_0)$ *be a strongly connected, live MG,* $Q \subseteq P$ *be a cut of* $\mathcal{N}$ *implying an extended system ES, two low level systems* LS_1,LS_2 *and a high level system HS.*

1. $L(ES) = L(\mathcal{N}, M_0)$
2. $R(ES)|_{P_N} = R(\mathcal{N}, M_0)$
3. $L(ES)|_{T_{LS_i}} = L(LS_i)$ *for* $i, j \in \{1,2\}, i \neq j$
4. $R(ES)|_{P_{LS_i}} = R(LS_i)$ *for* $i, j \in \{1,2\}, i \neq j$
5. $L(ES)|_{T_{HS}} = L(HS)$
6. $R(ES)|_{P_{HS}} = R(HS)$

Proof. 1. and 2. by definition of TT-MSIP and choice of initial marking according to Theorem 2
3. The argumentation given in [3] can be directly applied:
$L(ES)|_{T_{LS_i}} \subseteq L(LS_i)$: All sequences firable in ES are also firable in LS_i after removing the transitions of T_{A_2}. This is because we have removed all firing constraints appearing in ES imposed by $\mathcal{N}_{A_2}$.
$L(LS_i) \subseteq L(ES)|_{T_{LS_i}}$: We prove this part by contradiction. Let σ be a sequence of $L(LS_i)$ for which there is no $\sigma' \in L(ES)|_{T_{LS_i}}$ such that $\sigma = \sigma'|_{T_{LS_i}}$. Let σ_0 be the maximal prefix of σ for which there is a sequence $\sigma_0' \in L(ES)|_{T_{LS_i}}$ verifying $\sigma_0 = \sigma_0'|_{T_{LS_i}}$. If

[1] This is the main difference to the basic skeleton in [3], which contains the elements of the cut.

Proceedings of the Sixth International Workshop on Petri Nets and Performance Models (PNPM '95)
1063-6714/95 $10.00 © 1995 IEEE

$M_0^{ES}[\sigma_0' > M^{ES}$ and $M_0^{LS_i}[\sigma_0 > M^{LS_i}$, then trivially $\forall p \in P_{LS_i} : M^{ES}[p] = M^{LS_i}[p]$. The next transition to σ_0, t, in σ must be an output transition of a sink place of $\mathcal{N}_{A_j}$, because these transitions are the unique transitions of LS_i with additional constraints to fire in ES. Since places IP_{A_j} are implicit, these constraints must arise from $\mathcal{N}_{A_j}$. All maximal firable sequences in (EN, M) containing only transitions of $\mathcal{N}_{A_j}$ never can enable transition t because σ_0 is the maximal prefix of σ for which there is a sequence $\sigma_0' \in L(ES)|_{T_{LS_i}}$ verifying $\sigma_0 = \sigma_0'|_{T_{LS_i}}$. Let M' be a marking reachable in (EN, M) firing a maximal sequence σ_1 containing only transitions in $\mathcal{N}_{A_j}$. At M' all transitions of N_{A_j} are not enabled, hence have at least one empty input place. Moreover t has at least one empty input place being a sink place of $\mathcal{N}_{A_j}$ because t is not enabled at M'. Consequently an empty path from a source place to a sink place in $\mathcal{N}_{A_j}$ exists, which means that a place in IP_{A_j} corresponding to this path is an input place of t containing zero token. This contradicts the hypothesis that t is enabled in LS_i at M^{LS_i} reached by $M_0^{LS_i}[\sigma_0 > M^{LS_i}$.

4. To prove this observe that the set of relevant places coincide and according to 3. this equality holds.

5. and 6. follows from equivalent argumentation as for 3. and 4. since HS is in the same relation to LS_i as LS_i to ES. □

With this result the approximation algorithm given in [3] can be applied, i.e., a reduced set of TT-MSIPs is calculated by solving the all-pairs-shortest-path problem and the resulting high level and low level systems are solved successively to receive an approximate throughput as described by the Pelota algorithm in [3]. We propose in the following section an alternative method based on a decomposition of the underlying CTMC resulting directly from the decomposition of the SMG.

3 Matrix structures for an exact numerical analysis

Every marking $M = (M_1, M_2) \in R(HS)$ can be decomposed into the components M_1, including the marking of TT-MSIPs IP_{A_1} belonging to LS_1 and M_2, including the marking of TT-MSIPs IP_{A_2} belonging to LS_2. Let n_{HS} be the number of markings in $R(HS)$ and markings in $R(HS)$ are numbered consecutively from 1 to n_{HS}. This establishes a bijective mapping $index : R(HS) \longrightarrow \{1, \ldots, n_{HS}\}$, such that a marking from $R(HS)$ can be uniquely related to its number and vice versa, so numbers and markings are used interchangeable in the sequel. Define a $n_{HS} \times n_{HS}$ matrix $\mathbf{Q}_{HS}$ with $\mathbf{Q}_{HS}(x, y) = t$ for $t \in T_{HS}$, if t is enabled in marking x and the firing yields a new marking y and 0 otherwise. Since we consider marked graphs, if $x[t > y$, then no $t' \neq t$ with $x[t' > y$ can exist. Observe that $\mathbf{Q}_{HS}(x, x) = 0$ and $\mathbf{Q}_{HS}(x, y) = t$ implies $\mathbf{Q}_{HS}(x, z) \neq t$ for all z.

In the example $\mathbf{Q}_{HS}$ results from the net shown in Fig. 2.c) and contains as non-zero elements the indices of the transitions IT_i.

The reachability set of low level system i can be decomposed into disjoint subsets according to the marking of the TT-MSIPs, i.e., the corresponding marking of the high level system. So, let

$$R_z(LS_i) := \{M' \in R(LS_i) | M'|_{P_{HS}} = M \wedge index(M) = z\}$$

for $z \in \{1, \ldots, n_{HS}\}$ be the set of markings of LS_i with marking z on the places IP_{A_i}. Let $n_i := |R(LS_i)|$ be the number of markings in $R(LS_i)$ and denote by $n_i(z) := |R_z(LS_i)|$ the number of markings in $R_z(LS_i)$. As above assume that markings are numbered consecutively with respect to the markings of the high level system they belong to. Thus all markings from $R_1(LS_i)$ are followed by the markings from $R_2(LS_i)$ and so on.

For a low level system i we describe a set of matrices which all together describe the transition rates on the underlying CTMC. The firing of a transition from $T_i^{in} \cup T_i^{out}$ in a marking $M \in R_z(LS_i)$ uniquely determines the successor marking $z' \in R(HS)$. For matrix generation the rates of transitions from T_i^{in} are set to 1.0 and we assume single server semantic, which implies that the corresponding values can be interpreted as conditional probabilities of a transition on state space level after arrival of a token output bag in the low level system. For low level system i we generate two different sorts of matrices, active matrices describing transitions originated in i and passive matrices describing the reaction of the low level system after an arrival. Let $\mathbf{Q}[t, z, z']$ for $t \in T_i^{out} \cup \{\tau\}$ be a matrix including all transition rates related to transition t starting in a marking from $R_z(LS_i)$ and ending in a marking from $R_{z'}(LS_i)$. The situation $t = \tau$ covers all transitions in T_{A_i}. $\mathbf{Q}[\tau, z, z]$ is a $n_i(z) \times n_i(z)$ matrix since the successor marking has to be also in $R_z(LS_i)$, when firing a transition $t \notin T_i^{in} \cup T_i^{out}$ in some marking from $R_z(LS_i)$. $\mathbf{Q}[t, z, z']$ for $t \in T_i^{out}$ is a $n_i(z) \times n_i(z')$ matrix for $\mathbf{Q}_{HS}(z, z') = t$. The diagonal elements of $\mathbf{Q}_i[\tau, z, z]$ are defined as

$$\mathbf{Q}_i[\tau, z, z](x, x) = - \sum_{t \in T_i^{out} \cup \{\tau\}} \sum_{z' \in R(HS)} \sum_{y \in R_{z'}(LS_i)} \mathbf{Q}_i[t, z, z'](x, y) .$$

In a similar way can define for $t \in T_i^{in}$ matrices $\mathbf{U}_i[t, z, z']$ describing transitions originated in a marking from $R_z(LS_i)$ and ending in a marking from $R_{z'}(LS_i)$ for $\mathbf{Q}_{HS}(z, z') = t$, which implies that $\mathbf{U}_i[t, z, z']$ is a $n_i(z) \times n_i(z')$ matrix. It is easy to show that every row of $\mathbf{U}_i[t, z, z']$ contains only a single element with a value equal to 1, all other elements have zero values. For the sake of completeness we define $\mathbf{Q}_i[t, z, z'] = \mathbf{U}_i[t, z, z'] = \mathbf{0}$ for $\mathbf{Q}_{HS}(z, z') \neq t$.

Proceedings of the Sixth International Workshop on Petri Nets and Performance Models (PNPM '95)
1063-6714/95 $10.00 © 1995 IEEE

For LS_1 shown in Fig. 2.a) matrices $\mathbf{Q}[..]$ are generated for τ, covering the firing of transitions $T_1, \ldots, T_3$, for T_4 and for T_5. Matrices $\mathbf{U}[..]$ are generated for T_8 corresponding to IT_3 in the specification of LS_1. For LS_2 shown in Fig. 2.b) matrices $\mathbf{Q}[..]$ are generated for τ, covering T_6, T_7, and for T_8. Matrices $\mathbf{U}[..]$ are required for T_4 corresponding to IT_1 and T_5 corresponding to IT_2.

In the following we want to generate the reachability set $R(N, M_0)$ in a compositional way from the reachability sets of the high level system and the low level systems. To do so we make use of the well-known fact that the reachability problem for a live MG can be decided by considering its fundamental circuits, i.e. all reachable markings agree with the initial marking on the algebraic sum of tokens on all fundamental circuits. Note that a fundamental circuit f can contain paths in inverse directions, i.e. some coefficients in its vectorial description b_f might be negative.

Theorem 4 ([10]) *In a live marked graph (N, M_0), M_d is reachable from M_0 iff $B_f M_0 = B_f M_d$, where B_f is the fundamental circuit matrix.*

Theorem 5 *The set $R(ES)$ can be characterized as*

$$R(ES) = \cup_{z \in R(HS)} R_z(ES) = \cup_{z \in R(HS)} z \times R_z(LS_1)|_{P_{A_1}} \times R_z(LS_2)|_{P_{A_2}}$$

Proof. $R(ES) = \cup_{z \in R(HS)} R_z(ES)$ follows directly from the definition of $R_z(ES)$. It remains to show that for an arbitrary but fixed $z \in R(HS)$:
$R_z(ES) = z \times R_z(LS_1)|_{P_{A_1}} \times R_z(LS_2)|_{P_{A_2}}$
We prove first $R_z(ES) \subseteq z \times R_z(LS_1)|_{P_{A_1}} \times R_z(LS_2)|_{P_{A_2}}$.
Obviously $z \times R_z(LS_i)|_{P_{A_i}} = R_z(LS_i)$ holds. Furthermore the general result

$$R(ES)|_{P_1 \cup P_2} \subseteq R(ES)|_{P_1} \times R(ES)|_{P_2}$$

holds for arbitrary disjoint sets of places P_1 and P_2. Applying the results from Theorem 3 we get

$$\begin{aligned} R_z(ES) &= R_z(ES)|_{P_{HS} \cup P_{A_1} \cup P_{A_2}} \\ &\subseteq R_z(ES)|_{P_{HS}} \times R_z(ES)|_{P_{A_1}} \times R_z(ES)|_{P_{A_2}} \\ &= z \times R_z(ES)|_{P_{A_1}} \times R_z(ES)|_{P_{A_2}} \end{aligned}$$

It remains to show that $R_z(ES) \supseteq z \times R_z(LS_1)|_{P_{A_1}} \times R_z(LS_2)|_{P_{A_2}}$
Assume the contrary: let $M_d = (z, a_1, a_2)$ be such that $z \in R(HS), (z, a_1) \in R_z(LS_1)$ and $(z, a_2) \in R_z(LS_2)$ but $M_d \notin R_z(ES)$. According to Theorem 4 exists at least one fundamental circuit C with $b_C M_0^{ES} \neq b_C M_d$, where b_C is a fundamental circuit vector corresponding to C. Note that C is fundamental, i.e. it does not contain a circuit as a real subset, but it is not necessarily directed, i.e. it can contain paths which are reverse to the arc direction in N resulting in negative coefficients in b_C. Since $M_0^{LS_i} = M_0^{ES}|_{P_{LN_i}}$ and $R(LS_i) = RS(ES)|_{P_{LS_i}}$ C cannot be completely in LN_i, $i \in \{1, 2\}$. We will use this fact to derive the desired contradiction to our assumption. To get there we will transform C step by step into a circuit C_d such that C_d will be completely in LS_1 and the transformation will ensure that $b_{C_d} M_0^{ES} \neq b_{C_d} M_d$ holds iff $b_C M_0^{ES} \neq b_C M_d$.
We partition C into two sets of paths Π_1, Π_2 such Π_1 contains the paths which are in LN_1 and Π_2 contains the paths left in C. The paths in Π_1, Π_2 are chosen to be of maximal length, such that C is the concatenation of paths taken alternately from Π_1, Π_2. Obviously any $\pi \in \Pi_2$ is a path with a corresponding TT-MSIP x in EN (by definition of EN). Hence π together with x forms a fundamental circuit C' in EN. C' is completely in LN_2, so by the same argumentation as above $b_{C'} M_0^{ES} = b_{C'} M_d$ must hold for the corresponding circuit vector $b_{C'}$ since (z, a_2) is reachable in LS_2. If we exchange path π in C by x we yield another fundamental circuit C" which has a corresponding vector $b_{C''} = b_C - b_{C'}$. Now we get $b_C M_d = (b_{C''} + b_{C'}) M_d = b_{C''} M_d + b_{C'} M_0^{ES}$, which means that we can regard C" instead of C to argue that M_d is not reachable. This argumentation can be applied stepwise and consecutively for every element of Π_2, such that finally all elements of Π_2 are exchanged with their corresponding TT-MSIPs and the resulting circuit C_d is completely in LN_1 and $b_{C_d} M_0^{ES} \neq b_{C_d} M_d$. But this contradicts $(z, a_1) \in R_z(LS_1)$. □

The previous theorem allows the representation of $R(ES)$ using $R(HS)$ and $R(LS_i)$. The reachability set of the complete net can be completely characterized knowing the local reachability sets $R(LS_i)$ and $R(HS)$. For non-trivial low level systems the reachability sets of the low level systems and the high level system are much smaller than the reachability set of the complete SMG. However, only the former need to be generated and stored. For the number of markings in the different sets we get the following relation.

$$n = \sum_{z=1}^{n_{HS}} n_1(z) n_2(z)$$

In a similar way the generator matrix of the CTMC underlying the complete SMG can be computed. The state space of the CTMC is isomorphic to $R(ES)$, if we assume exponentially distributed firing times for all transitions, which is the case here. Instead of the cross-product of subspaces we now use tensor (kronecker) products and sums [2, 5, 11] to combine submatrices related to low level systems. Let $\mathbf{Q}$ be the $n \times n$ generator matrix and let $\mathbf{Q}[z, z']$ be the $n(z) \times n(z')$ submatrix describing transitions between markings from $R_z(ES)$ and $R_{z'}(ES)$. $\mathbf{Q}$ is block-structured in these submatrices and to every block $\mathbf{Q}[z, z']$ belongs one element $\mathbf{Q}_{HS}(z, z')$. The different blocks can be computed from the matrices of the

Proceedings of the Sixth International Workshop on Petri Nets and Performance Models (PNPM '95)
1063-6714/95 $10.00 © 1995 IEEE

low level systems as follows.

$$\mathbf{Q}[z,z'] = \begin{cases} \mathbf{Q}_1[\tau,z,z] \oplus \mathbf{Q}_2[\tau,z,z] & \text{if } z = z' \\ \mathbf{Q}_1[t,z,z'] \otimes \mathbf{U}_2[t,z,z'] & \text{if } \mathbf{Q}_{HS}(z,z') = t \text{ and } t \in T_1^{out} \\ \mathbf{U}_1[t,z,z'] \otimes \mathbf{Q}_2[t,z,z'] & \text{if } \mathbf{Q}_{HS}(z,z') = t \text{ and } t \in T_2^{out} \\ \mathbf{0} & \text{otherwise} \end{cases} \quad (1)$$

The above representation can be used to generate $\mathbf{Q}$ efficiently and it can be directly exploited in iterative numerical solution techniques for stationary and transient analysis of the underlying CTMC. The stationary or transient distribution of the CTMC is computed with an established iterative solution technique, but without storing or generating matrix $\mathbf{Q}$, corresponding algorithms are proposed in [2, 12, 13]. This analysis approach allows the analysis of models with a very large state space, much larger (i.e. by an order of a magnitude) than solvable with conventional methods. But the method is still exact, in the framework of numerical analysis, and introduces no approximation error. However, realistic models often yield extremely huge state spaces, which require long solution times or cannot be solved at all, even with the tensor based solution techniques. For these models approximate solution techniques have to be developed.

4 Approximate solution methods in the framework of tensor based matrix structures

We propose here an approach based on the above matrix representation which coincides with other known approximations [3, 4, 7, 8] and gives therefore a theoretical underpinning of these approaches. Additionally, the integration of the approximation in the above matrix representation enables us to compute some estimate for the approximation error, which is usually not possible with other known techniques. If the errors of the approximation are too large and the state space of the model is in the range of a few million states, then the approximation can be used as initial guess for an iterative solution technique based on the tensor representation. This two level solution often reduces the time of an exact analysis technique drastically compared with the standard method starting with an arbitrary initial distribution for iterative numerical analysis.

4.1 A general approximation scheme

The general idea of the approximation techniques can be summarized in the following steps for a single cut. The extension to more than two low level systems is straight forward (see also the remarks at the end of this paper). Other approximation methods known from literature are usually based on the steps 1 to 6 in the proposed or a similar form.

Decompose the MG in LS_1, LS_2 and HS (step 1)
Compute initial aggregates for LS_i (step 2)
REPEAT
  Analyze LS_i, where LS_j is represented by the aggregate $(i = 1, 2)$ (step 3)
  Compute new aggregate parameters (step 4)
  Solve HS, where LS_i is represented by the coresp. aggregate or estimate the solution (step 5)
UNTIL convergence of the solution (step 6)
Estimate the approximation error (step 7)
IF approximation error too large THEN
  Perform iterative solution (step 8)
Compute required performance quantities (step 9)

We now describe the different steps of the approach. The decomposition of the net performed in step 1 has already been introduced, iterative solution based on the tensor representation, as required in step 8, have been mentioned above. The remaining seven steps are explained in the following. Naturally different possibilities exist to define these steps and we consequently present some alternatives.

4.2 Subnet analysis and aggregate parameter computation

First the aggregate type for a low level system has to be fixed, we use here an aggregate which reduces LS_i to the set T_i^{out}. So, every $t \in T_i^{out}$ describes an exponential transition with state independent or state dependent transition rate. If the transition rate is state dependent, then it is allowed to depend on the marking of the places in IP. With this approach aggregates can be integrated directly in the high level system and the low level systems by simply assigning transition rates to the corresponding transitions. In fact, the aggregate for LS_i is part of the specification of LS_j, only transition rates have to be added. HS includes, up to the transition rates, aggregates for LS_1 and LS_2. Denote by λ_t the transition rate of transition $t \in T_i$ which might depend on the current marking, however, marking dependency is restricted to markings of places in the same low level system and to non-zero values for all markings where t is enabled. In LS_i the transition rates for transitions from T_i^{in}, which represent an aggregate for LS_j, are denoted as $\mu_t(z)$, where $t \in T_i^{in}$ and $z \in \{1, \ldots, n_{HS}\}$ describes the marking of the places in IP.

To perform step 2 the values $\mu_t(z)$ can be set to some values μ_t which is a guess for the aggregate parameters. Alternatively one can compute aggregate parameters by pre-analysis, which means to solve the following set of equations.

$$\mathbf{y}_i[z]\mathbf{Q}_i[\tau,z,z] + \sum_{t \in T_i^{out}} \mathbf{Q}_i[t,z,z']\mathbf{G}_i[z',z] = \mathbf{0} \quad (2)$$

Proceedings of the Sixth International Workshop on Petri Nets and Performance Models (PNPM '95)
1063-6714/95 $10.00 © 1995 IEEE

and $\mathbf{y}_i[z]\mathbf{e}^T = 1.0$, where $z[t > z'$, $\mathbf{e}^T$ denotes the unit-vector and $\mathbf{G}_i[z, z']$ is a non-negative $n_i(z) \times n_i(z')$ matrix with unit row sum.

The choice of the **G**-matrices is in principle arbitrary, sometimes they might be realized by a corresponding **U**-matrices, if they exist, such that the equations describe the short-circuited low level system. From the vector $\mathbf{y}_i[z]$ the aggregate parameters can be computed as

$$\mu_t(z) = \mathbf{y}_i[z]\mathbf{Q}[t, z, z']\mathbf{e}^T \ . \qquad (3)$$

To perform step 3, the analysis of low level system i combined with an aggregate for system j, the above aggregate parameters can be used. The generator matrix $\mathbf{Q}^i$ of the resulting system can be easily derived from equation (1). $\mathbf{Q}^i$ is, like $\mathbf{Q}$, block-structured corresponding to the the marking of the places IP. Block $\mathbf{Q}^i[z, z']$ is computed with the same formulas than $\mathbf{Q}[z, z']$ except all matrices belonging to LS_j are substituted by the corresponding aggregate parameters. Thus, a non-zero matrix $\mathbf{U}_j[t, z, z']$ is substituted by 1.0 and a matrix $\mathbf{Q}_j[t, z, z']$ is substituted by $\mu_t(z)$. Observe that the number of markings in $R_z(ES)$ is reduced from $n_1(z)n_2(z)$ to $n_1(z)$ or $n_2(z)$ by this aggregation step. Apart from data structures for the aggregate parameters, no additional data structures are required to represent $\mathbf{Q}^1$ and $\mathbf{Q}^2$, when $\mathbf{Q}$ is represented exploiting the tensor structure. Analysis of LS_i means the solution of the system

$$\mathbf{p}^i\mathbf{Q}^i = \mathbf{0} \text{ and } \mathbf{p}^i\mathbf{e}^T = 1.0 \ . \qquad (4)$$

Each vector $\mathbf{p}^i$ can be decomposed into subvectors $\mathbf{p}^i[z]$ including probabilities for states from $R_z(LS_i)$, due to the ordering of states $\mathbf{p}^i = (\mathbf{p}^i[1], \ldots, \mathbf{p}^i[n_{HS}])$. Define $\bar{\mathbf{p}}^i[z]$ as $\mathbf{p}^i[z]/(\mathbf{p}^i[z]\mathbf{e}^T)$, i.e. the subvector is normalized to 1. New aggregate parameters, in step 4, are computed from $\mathbf{p}^i$ as

$$\mu_t(z) = \bar{\mathbf{p}}^i[z]\mathbf{Q}_i[t, z, z']\mathbf{e}^T \ , \qquad (5)$$

which yields an aggregate with marking dependent parameters, similar to the approach chosen in [7]. The aggregate parameters for LS_j reflect the conditional throughputs of $t \in T_i^{in}$ with marking z on the places IP. In [3, 8] aggregates with state independent parameters, approximating the mean delay of a low level system have been introduced. Corresponding aggregate parameters can be computed as follows.

$$\mu_t(z) = \mu_t = \frac{\sum_{z=1}^{n_{HS}} \mathbf{p}^i[z]\mathbf{Q}_i[t, z, z']\mathbf{e}^T}{\sum_{z=1, z[t>}^{n_{HS}} \mathbf{p}^i[z]\mathbf{e}^T} \qquad (6)$$

In this case aggregate parameters reflect the mean throughput of the corresponding transition.

Observe that (5) and (6) are not identical to the corresponding computations in [3, 7, 8]. With the detailed knowledge of conditional distributions our methods of parameter computations seem to be more natural and efficient. The alternative approaches proposed in the mentioned papers compute aggregate parameters in a more complex way using net-level results. However, this introduces additional effort and, possibly, convergence problems. Nevertheless, other forms of aggregate parameter computation can be integrated in our framework and do not affect the general approach. At the current stage it is not possible to decide in general which form of parameter computation gives the best results, although examples indicate that state dependent rates yield significantly smaller errors without introducing additional effort or convergence problems (see also the example in Sec. 5). This result has also been observed in a recent paper analyzing a more general class of nets with a similar technique [9].

Step 5 is optional but needs not much effort, it is required if the overall solution should be approximated by means of the solutions $\mathbf{p}^i$ and it might also be useful in improving the approximation as proposed in [3]. The generator matrix of the HS, where LS_1 and LS_2 are substituted by their aggregates has the same structure as $\mathbf{Q}_{HS}$. The matrix elements can be computed from (1) after substituting all non-zero **U**-matrices by 1.0 and all matrices $\mathbf{Q}_i[t, z, z']$ by $\mu_t(z)$. We denote the resulting matrix by $\mathbf{Q}^0$, since it can be directly derived from (1), no additional data structures are required for this matrix. The stationary solution $\mathbf{p}^0$ for this system is computed via (4) setting $i = 0$. From $\mathbf{p}^0$ the throughput of HS can be computed and compared with the throughputs determined from $\mathbf{p}^1$ and $\mathbf{p}^2$, if the values differ, aggregate parameters might be scaled as outlined in [3].

The convergence of the approach, which is checked in step 6, is a crucial point. First of all, neither [3, 4, 7, 8] nor us can guarantee that the specific approximate method converges and even if it does, the quality of the gained approximate results is unknown. However, experience shows that usually only a few iterations are required before convergence is reached. Convergence is assumed if the parameters computed for the aggregates or the solution vectors for the aggregated systems differ only by a small factor in consecutive iterations. Let $\mathbf{p}^{i\,(k)}$ and $\mu_t^{(k)}$ be the solution vector and aggregate parameter computed in the k-th iteration of the algorithm. Convergence according to the solution vector is assumed when

$$\|\mathbf{p}^{i\,(k)} - \mathbf{p}^{i\,(k-1)}\| \ \leq \ \epsilon_1 \text{ and } \qquad (7)$$

$$\sum_{z=1}^{n_{HS}} |\mathbf{p}^{1\,(k)}[z]\mathbf{e}^T - \mathbf{p}^{2\,(k)}[z]\mathbf{e}^T| \ \leq \ \epsilon_2 \qquad (8)$$

holds. Convergence according to the aggregate parameters is given when

$$\frac{\mu_t^{(k)}(z) - \mu_t^{(k-1)}(z)}{\mu_t^{(k-1)}(z)} \leq \epsilon \text{ for all } z \in \{1, \ldots, n_{HS}\}$$

Proceedings of the Sixth International Workshop on Petri Nets and Performance Models (PNPM '95)
1063-6714/95 $10.00 © 1995 IEEE

is observed. Which of both approaches is preferable cannot be decided at the current stage. Thus, both criterions might be checked.

4.3 Error estimation and result improvement

Once the iteration has terminated, the quality of the results is usually unknown, even if convergence has been observed. This is one of the main drawbacks of most approximate solution techniques. However, it is important to get at least an estimate of the error. The solution of the complete model can be approximated via

$$\tilde{\mathbf{p}}[z] = \mathbf{p}^0(z)\tilde{\mathbf{p}}^1[z] \otimes \tilde{\mathbf{p}}^2[z] \qquad (9)$$

Let $\mathbf{p}$ be the stationary solution of the complete system (i.e. $\mathbf{pQ} = \mathbf{0}$ and $\mathbf{pe}^T = 1.0$). The error of the approximation is described by $||\mathbf{p} - \tilde{\mathbf{p}}||$, however, since $\mathbf{p}$ is unknown it cannot be computed exactly. An alternative is to check the residuals $||\tilde{\mathbf{p}}\mathbf{Q}||$ as an error estimate, which usually works fine, although we cannot guarantee that small residuals imply a good approximation [13]. Additionally, a bad approximation of the solution vector does not imply that performance quantities like throughputs are also bad approximations. However, the computation of residuals is much better than nothing and helps in almost all cases to decide whether the approximation meets the required accuracy. It should be noticed that the residuals can be computed without generating $\mathbf{Q}$ by exploitation of the tensor structure, and they can even be computed without a memory expensive representation of $\tilde{\mathbf{p}}$ since elements of $\tilde{\mathbf{p}}$ can be generated from the vectors of the aggregated systems on demand.

If the residuals indicate a large error, some iteration steps should be performed starting with $\tilde{\mathbf{p}}$. This is possible by the tensor based approach for all systems, for which we are able to store the isolated matrices plus two vectors of size n. On current workstations this works well for state spaces with several millions of states.

Performance quantities of the system, as required in step 8, are usually defined locally in LS_i or HS. Thus they are computed directly from the vector $\mathbf{p}^i$ ($i = 0, 1, 2$) or, if $\tilde{\mathbf{p}}$ has been improved by an iterative technique, by mapping $\tilde{\mathbf{p}}$ on $\mathbf{p}^i$, which is done by appropriately summing of vector components.

5 Example

We use here a very simple example, which, however, is sufficient to show the benefits of the new approach. The different parts of the example are shown in Figs. 1 and 2. Implicit places from IP are shaded, the extended system ES in Fig. 1 b) results from the original net a) by adding implicit places. The cut Q contains the places P_1, P_8 and P_9. We assume that all transitions have single server semantics and, apart from T_2, T_3, T_6 and T_7, have transition rate 1.0. Transitions T_2 and T_6 have rate λ, transitions T_3 and T_7 rate μ. The initial markings for the different nets are shown in the graphical representation, K is an integer parameter which is modified to generate reachability sets of different sizes.

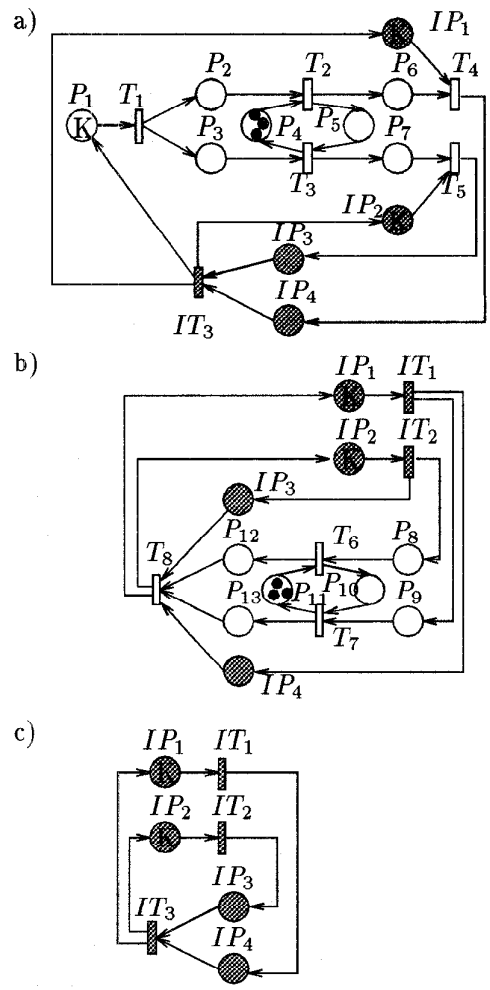

Figure 2: a) low level system LS_1, b) low level system LS_2, and c) high level system HS for example SMG

In a first step the marking sets and matrices for the low level systems and the high level system are computed. In Tab. 1 the sizes of the different marking sets and the number of non-zero elements in the different matrices are shown for different values of K. The number of states grows rapidly if the value of K is increased. The number of non-zero elements in $\mathbf{Q}$ ($\#nz(\mathbf{Q})$), which is shown in the third column, grows even faster. For a numerical analysis using standard techniques and sparse storage of $\mathbf{Q}$ the required space is proportional to $2n_{ES} + \#nz(\mathbf{Q})$. If the tensor structure is exploited in an iterative solution, then the required storage is proportional to $2n_{ES} + \#nz(\mathbf{Q}_i, \mathbf{U}_i)$,

Proceedings of the Sixth International Workshop on Petri Nets and Performance Models (PNPM '95)
1063-6714/95 $10.00 © 1995 IEEE

K	n_{ES}	$\#nz(\mathbf{Q})$	n_{HS}	n_{LS_1}	n_{LS_2}	$\#nz(\mathbf{Q}_i, \mathbf{U}_i)$
3	228	876	16	98	64	635
5	1704	8224	36	476	243	2995
10	38476	234830	121	5201	1633	35586
15	262348	1486488	256	22926	5173	155721

Table 1: Reachability set and matrix size of the example

where the latter describes the number of non-zero elements in the isolated matrices required to represent the generator. For large reachability sets the number of states exceeds the number of non-zeros in the isolated matrices even if a net is cut only in two parts. Simple comparison of storage requirements shows that the size of state spaces solvable by the structured approach is about one order of magnitude larger than the size of state spaces solvable by the conventional approach. Additionally, state space and matrix generation is faster in the structured approach, since the generation of a few small state spaces is much more efficient than the generation of one large state space.

Table 2 compares the quality of results obtained from different approximate analysis techniques for the example SMG. In all techniques the iteration stops when the normalized difference of the transition rates for the aggregates is less than $\epsilon = 0.0001$. This turns out to allow a very fast solution with every chosen technique: the number of iterations is 3 or 4.

Three particular performance measures are considered, namely the throughput, which is identical for all transitions in the net, the mean population on place IP_3 and the mean population on place P_{11}. The former two results are computed from HS, the latter from LS_2. Table 2 gives the relative errors for the mentioned quantities; errors smaller than 0.01% are denoted as 0.0%. The results show that the state dependent aggregate is superior to the aggregate with fixed transition rates. This is caused by the structure of the example, which contains a serialization in firing transitions T_2 and T_3 or T_6 and T_7. Aggregates with fixed transition rates do no reflect this behavior adequately. The results for the state dependent aggregate are nearly exact and probably sufficiently accurate for most applications. The values of the maximum norm of the vector of residuals gives a fairly accurate estimate of the quality of results, although we cannot expect a linear dependency between the maximum norm of the residuals and the errors in results like mean throughput or population.

In Tab. 3 we compare the convergence of the power method starting with different initial distributions. We compare the exact solution $\mathbf{p}$ with the solution reached after *iter* iteration steps. The initial distribution $\mathbf{e}_1$ assigns probability 1.0 to the initial state/marking and probability 0.0 to all other states. This initial distribution is often used in tensor based approaches for the analysis of stochastic automata based models [6, 11]. However in this example, it gives

iter	$\mathbf{e}_1$	$1/n\mathbf{e}$	$\tilde{\mathbf{p}}_1$	$\tilde{\mathbf{p}}_2$
0	9.91e-01	7.07e-03	2.54e-02	4.31e-04
10	7.10e-02	5.73e-03	5.97e-03	1.02e-04
50	2.36e-03	4.96e-04	4.63e-04	1.17e-06
100	1.02e-04	2.17e-05	2.03e-05	5.03e-08
300	3.56e-10	8.01e-11	7.48e-11	1.76e-13

Table 3: $||\mathbf{p} - \mathbf{p}^{(iter)}||_\infty$ for different initial distributions.

a bad convergence. The uniform distribution denoted by $1/n\mathbf{e}$ performs much better. The values in column $\tilde{\mathbf{p}}_1$ are obtained with the approximate solution calculated by the aggregation approach with state independent aggregates. This distribution is not better than the uniform distribution, although performance quantities resulting from the direct use vector $1/n\mathbf{e}$ as an approximation of $\mathbf{p}$ yield much larger errors than for $\tilde{\mathbf{p}}_1$. $\tilde{\mathbf{p}}_2$ is the distribution resulting from the approximation with state dependent aggregates, it is obviously the best choice of an initial distribution for this example. The initial distribution has a significant effect on the number of iterations necessary to reach a required accuracy for the solution. For an estimated accuracy of 10^{-5}, 160 iterations are needed starting with $\mathbf{e}_1$, with $1/n\mathbf{e}$ and $\tilde{\mathbf{p}}_1$ 130 iterations are required and starting with $\tilde{\mathbf{p}}_2$ only 40 iterations are necessary. Since approximate solutions can be computed for large models very efficiently compared to the solution time of the complete model, a combination of approximate and exact analysis will often yield a significant reduction of the overall solution time.

6 Conclusions

In this paper we describe an exact iterative numerical technique and an approximation technique to analyze Markov chains described by stochastic marked graph nets. The techniques are based on a structured description of the generator matrix $\mathbf{Q}$, such that that submatrices are described by tensor sums and products of relatively small matrices. We prove that the structured description not just describes a superset of $\mathbf{Q}$ – which is often the case in tensor based approaches – our structured description exactly matches $\mathbf{Q}$. This allows to choose freely an initial distribution for the iterative numerical technique, e.g. one derived from an approximate technique. The memory efficient representation of $\mathbf{Q}$ enlarges the size of solvable models by about one order of magnitude compared to conventional methods.

Proceedings of the Sixth International Workshop on Petri Nets and Performance Models (PNPM '95)
1063-6714/95 $10.00 © 1995 IEEE

		fixed trans. rates				
μ	λ	err. thr.	err. $\#IP_3$	err. $\#P_{11}$	$\|\bar{\mathbf{p}}\mathbf{Q}\|_\infty$	# iter
10.0	0.1	-4.33%	+5.60%	-4.33%	5.36e-3	4
1.0	1.0	-12.5%	+14.1%	-10.10%	1.85e-2	4
0.1	10.0	-2.36%	+0.01%	+0.17%	8.98e-3	3
		state dep. trans. rates				
0.1	10.0	+0.00%	+0.00%	+0.00%	4.52e-5	3
1.0	1.0	+0.14%	-0.02%	+0.01%	7.64e-4	3
10.0	0.1	+0.00%	+0.00%	+0.00%	7.33e-5	3

Table 2: Approximation results for the example with $K = 3$.

From a theoretical point of view this new structured description gives a nice notational framework to classify and compare well-known approximate techniques as proposed by [3, 4, 7, 8]. The practical implications are twofold: The quality of the approximate results can be checked for fairly large state spaces by analyzing the vectors of residuals and exact solution techniques can be accelerated by using approximate results as initial distribution for exact iterative solution techniques.

The example given above nicely demonstrates how the numerical analysis of SMGs profits from our new technique.

Generalizing our approach from a single, simple cut to a partition of a SMG into more than two parts is straightforward, since the parts are SMGs again. This generalization is left out here due to lack of space, but we recommend to take it into consideration for practical applications. One decision to make is where to cut a SMG. First, cuts can be chosen according to the physical structure of the modeled system. Second, cuts can be made to support the solution. In the latter case subnets should be chosen such that their marking sets are approximately of the same size, which allows a representation of $\mathbf{Q}$ with minimum storage requirements.

References

[1] C. Beounes. Stochastic Petri net modeling for dependability evaluation of complex computer systems. In: *Proc. Int. Workshop on Timed Petri nets Torino, Italy, IEEE Press (1985) 191-198.*

[2] P. Buchholz. A hierarchical view of GCSPNs and its impact on qualitative and quantitative analysis. *Journal of Parallel and Distributed Computing 15 (1992) 207-224.*

[3] J. Campos, J. M. Colom, H. Jungnitz, M. Silva. A general iterative technique for approximative throughput computation for stochastic marked graphs. In: *Proc. 5th Int. Workshop on Petri Nets and Performance Models, IEEE Press (1993) 138:147..*

[4] G. Ciardo, K. Trivedi. A decomposition approach for stochastic reward net models; *Performance Evaluation 18 (1994) 37-59.*

[5] M. Davio. Kronecker products and shuffle algebra. *IEEE Trans. on Comp. 30 (1981) 116-125.*

[6] S. Donatelli. Superposed Generalized Stochastic Petri nets: definition and efficient solution. In *R. Valette (ed.), Application and Theory of Petri Nets 1994, Springer LNCS 815 (1994) 258-277.*

[7] Y. Li, C. M. Woodside. Iterative decomposition and aggregation of stochastic marked graph Petri nets. In: *Proc. of the 12th Int. Conference on Application and Theory of Petri Nets (1991) 257-275.*

[8] Y. Li, C. M. Woodside. Performance Petri net analysis of communications protocol software by delay equivalent aggregation. In: *Proc. 4th Int. Workshop on Petri Nets and Performance Models, IEEE Press (1991) 64-73.*

[9] Y. Li, C. M. Woodside. Complete decomposition of stochastic Petri nets representing generalized service networks. *IEEE Trans. on Comp. 44 (1995) 577-592.*

[10] T. Murata. Petri nets: properties, analysis and applications. *Proc. of the IEEE 77 (1989) 541-580.*

[11] B. Plateau. On the stochastic structure of parallelism and synchronization models for distributed algorithms. In: *Proc. ACM Sigmetrics Conf. on Measurement and Modelling of Computer Systems (1985).*

[12] W.J. Stewart, K. Atif, B. Plateau. The numerical solution of stochastic automata networks. to appear in *European Journ. of Oper. Res.*

[13] W.J. Stewart. Introduction to the numerical solution of Markov chains. *Princeton University Press (1994).*

Proceedings of the Sixth International Workshop on Petri Nets and Performance Models (PNPM '95)
1063-6714/95 $10.00 © 1995 IEEE