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
A path of \(N = (P, T, F)\) is a sequence \(x_1 \ldots x_k\) with \(x_i \in P \cup T\) and \((x_{i-1}, x_i) \in F\) for \(1 \leq i \leq k\). 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 fullfills \(x_1 = x_k\) but it need not be a path itself, i.e., some paths inside the sequence might be in opposite direction. The set \(P(x, y)\) 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 \(P(X, Y) = \cup_{x \in X, y \in Y} P(x, y)\). A net is strongly connected if for every two elements \(x, y \in P \cup T\), then \(P(x, y) \neq \emptyset\). Pre- and post-sets of elements are denoted by \(x^* = \{y \mid (y, x) \in F\}\) and \(x^\bullet = \{y \mid (x, y) \in F\}\). 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 \((N, M_0)\) where \(M_0\) is the initial marking. Transition \(t \in T\) is enabled in marking \(M\), if \(M[p] > 0\) for all \(p \in t\). A transition \(t\) enabled in marking \(M\) can fire yielding a new marking \(M'\) with \(M'[p] = M[p] - \delta(p \in t)\), where \(\delta(x \in X) = 1\) for \(x \in X\) and 0 otherwise. This is denoted as \(M[t] > M'\). For a sequence \(\sigma = t_1 \ldots t_k\) with \(M_1[t_1] > M_2[t_2] > \ldots > M_k[t_k]\), we use the abbreviation \(M_1[\sigma] > M_k[\sigma]\), marking \(M_k[\sigma]\) is reachable from \(M_1\) by firing \(\sigma\). The reachability set \(R(N, M_0)\) 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: \(\exists \sigma : M[\sigma] > M' > M[t]\). The language \(L(N, M_0)\) of a net system is defined as the set of all firing sequences \(\sigma \in \min \{M[\sigma]\} \geq M_0[\sigma]\). An MSIP is a place added with one input transition and one output transition.

**Theorem 1** ([3]) Let \(N = (P, T, F)\) be a strongly connected MG and \(p \notin P\) be a place added with one input transition \(t_i \in T(p)\) and one output transition \(t_o \in T(p)\) such that \(y^o = t_o\). \(p\) is an MSIP in \(N\) iff there exists \(\gamma \geq 0\) such that \(y^o C = Lp\), where \(C\) is the incidence matrix of \(N\).

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

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

**Definition 1** Let \(N = (P, T, F)\) be a strongly connected MG. A subset of places \(Q \subseteq P\), is said to be a cut of \(N\) iff there exist two subnets, \(N_1 = (P_1, T_1, F_1)\) and \(N_2 = (P_2, T_2, F_2)\), of \(N\) verifying

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

\(Q^n_i := T_i \cap Q (Q^n_i := T_i \cap Q)\) for \(i \in \{1, 2\}\) is the set of input (output) places of subnet \(N_i, T^n_i := Q^n_i \cap T^n_i\) is the set of input (output) transitions for \(N_i\).

According to the above definition and the properties of MGs \(Q^n_1 \cap Q^n_2 = \emptyset\) and \(Q^n_1 \cup Q^n_2 = Q\). 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 \(p\) to \(p^\bullet\) circumventing \(p\). Such a path can be described by a vector \(Y \in \mathbb{F}\).

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

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

1. \(P_{A_i} = P \setminus Q^n_{i^\bullet}\)
2. \(T_{A_i} = T_i \setminus T^n_{i^\bullet}\)
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 \(p \cap T_{A_i} = \emptyset\) are source places and places \(p \in P_{A_i}\) with \(p \cap T_{A_i} = \emptyset\) are sink places of \(N_{A_i}\). \(P_{A_i}\) denotes the set of paths of \(N_{A_i}\), from a source place to a sink place. \(P_{A_i}\) is the set of TT-MSIPs which result from \(P_{A_i}\). For every \(\pi \in P_{A_i}\), we introduce a TT-MSIP \(p_{\pi} \in P_{A_i}\) given by the linear combinations of the rows in the incidence matrix corresponding to the places in \(\pi\). We denote

\(\text{IPA}_i = \{A_i \mid A_i \text{ is a TT-MSIP in } N_{A_i}\}\)

\(\text{IPA}_i = \{A_i \mid A_i \text{ is a TT-MSIP in } N_{A_i}\}\).
The initial marking is denoted as $M_0^p$ and is given by $M_0^p := \min\{\sum_{p \in \pi} M_0[p] \mid p = \sum_{p \in \pi} l_{p_i}, \pi \in \mathcal{P}_A\} \text{ if } p \in IP$ and $M_0^p := M_0[p]$ otherwise.

**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 $NA_i = (PA_i, TA_i, FA_i)$, $i \neq j$ from $ES$, i.e. $PLN_i := EN \setminus PA_i$, $T LN_i := EN \setminus TA_i$, $FLN_i := F EN \cap (PLN_i \times T LN_i) \cup (T LN_i \times PLN_i)$, and $M_0^{LN_i} := M_0^{EN} | PLN_i$.

**High level system** $HS = (HN, M_0^{HN})$: $HS$ is derived by subtracting $NA_i$ for $i \in \{1, 2\}$ 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 $N$ have only $Q$ in common, but do not share any places'. 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 $(N, M_0)$ be a strongly connected, live MG, $Q \subseteq P$ be a cut of $N$ implying an extended system $ES$, two low level systems $LS_i, LS_j$ and a high level system $HS$.

1. $L(ES) = L(N, M_0)$
2. $R(ES)|_{T_{LS_i}} = R(N, M_0)$
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 $TA_i$. This is because we have removed all firing constraints appearing in $ES$ imposed by $NA_i$.
   $L(LS_i) \subseteq L(ES)|_{T_{LS_i}}$: We prove this part by contradiction. Let $\sigma$ 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 $\sigma_0$ be the maximal prefix of $\sigma$ for which there is a sequence $\sigma'_0 \in L(ES)|_{T_{LS_i}}$ verifying $\sigma_0 = \sigma'_0 |_{T_{LS_i}}$. If

---

**Definition 3** Let $(N, M_0)$ be a strongly connected, live MG, $Q \subseteq P$ be a cut of $N$ yielding two subnets $N_1 = (P_1, T_1, F_1)$ and $N_2 = (P_2, T_2, F_2)$.
\[ M^{ES}[\sigma_0] > M^{ES} \] and \( M_0^{LS}[\sigma_0] > M^{LS} \), then trivially \( \forall p \in P_{LS} : M^{ES}\{p\} > M^{LS}\{p\} \). The next transition to \( \sigma_0 \), if in \( \sigma \) must be an output transition of a sink place of \( N_A \), because these transitions are the unique transitions of \( LS \) with additional constraints to fire in \( ES \). Since places \( IP_A \) are implicit, these constraints must arise from \( N_A \). All maximal firable sequences in \( (EN, M) \) containing only transitions of \( N_A \) never can enable transition \( t \) because \( \sigma_0 \) is the maximal prefix of \( \sigma \) for which there is a sequence \( \sigma_0' \in L(ES) \), verifying \( \sigma_0 = \sigma_0'[T_{LS}] \). Let \( M' \) be a marking reachable in \( (EN, M) \) firing a maximal sequence \( \sigma_1 \) containing only transitions in \( N_A \). At \( M' \) all transitions of \( N_A \) 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 \( N_A \), because \( t \) is not enabled at \( M' \). Consequently an empty path from a source place to a sink place in \( N_A \) exists, which means that a place in \( IP_A \) corresponding to this path is an input place of \( t \) containing zero token. This contradicts the hypothesis that \( t \) is enabled in \( LS \) at \( M^{LS} \) reached by \( M^{LS}[\sigma_0] > M^{LS} \).

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 \( LS \) as \( LS \) to \( ES \).

With this result the approximation algorithm given in [3] can be applied, i.e., a reduced set of \( TT-\text{MSIPs} \) 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 Pelotia 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 \( M_1 \) and \( M_2 \) belonging to \( LS \) and \( M_2 \), including the marking of \( TT-\text{MSIPs} \) \( IP_A \) belonging to \( LS \) and \( LS \). Let \( n_{HS} \) be the number of markings in \( R(HS) \) and \( n_{LS} \) in \( R(LS) \) are numbered consecutively from 1 to \( n_{HS} \). This establishes a bijective mapping \( \text{index} : R(HS) \to \{1, \ldots, n_{HS}\} \), such that a marking in \( R(HS) \) can be uniquely related to its number and vice versa, so numbers and markings are used interchangeably in the sequel. Define a \( n_{HS} \times n_{HS} \) matrix \( Q_{HS} \) with \( Q_{HS}(x,y) = t \) for \( t \in T_{HS} \), if \( y \) 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 \( t \neq t' \) with \( x[t'] = y \) can exist. Observe that \( Q_{HS}(x,y) = 0 \) and \( Q_{HS}(x,y) = t \) implies \( Q_{HS}(x,z) \neq t \) for all \( z \).

In the example \( Q_{HS} \) results from the net shown in Fig. 9. (c) and contains as non-zero elements the indices of the transitions \( T_i \).

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

\[ R_i(LS_i) := \{M' \in R(LS_i)|P_{HS} = M \wedge \text{index}(M) = z\} \]

for \( z \in \{1, \ldots, n_{HS}\} \) be the set of markings of \( LS_i \) with marking \( x \) on the places \( IP_A \). Let \( n_i := |R(LS_i)| \) be the number of markings in \( R(LS_i) \) and denote by \( n_i(z) := |R_i(LS_i)| \) the number of markings in \( R_i(LS_i) \). 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 \( R_i(LS_i) \) are followed by the markings from \( R_j(LS_j) \) 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 a marking \( M \in R(LS_i) \) uniquely determines the successor marking \( x' \in Q_{HS} \). For matrix generation the rates of transitions from \( T_i \) 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. For low level system \( i \) we generate a set of matrices \( Q_i \) 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. 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.
For LS\textsubscript{1} shown in Fig. 2.a matrices Q[.] are generated for φ, covering the firing of transitions T\textsubscript{1}, \ldots, T\textsubscript{3}, for T\textsubscript{4} and for T\textsubscript{5}. Matrices U[.] are generated for T\textsubscript{8} corresponding to IT\textsubscript{8} in the specification of LS\textsubscript{1}. For LS\textsubscript{2} shown in Fig. 2.b matrices Q[.] are generated for φ, covering T\textsubscript{8}, T\textsubscript{7}, and for T\textsubscript{4}. Matrices U[.] are required for T\textsubscript{8} corresponding to IT\textsubscript{1} and IT\textsubscript{2}.

In the following we want to generate the reachability set R(N, M\textsubscript{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\textsubscript{f} might be negative.

**Theorem 4 ([10])** In a live marked graph (N, M\textsubscript{0}), M\textsubscript{4} is reachable from M\textsubscript{0} iff B\textsubscript{T}\textsubscript{1} M\textsubscript{0} = B\textsubscript{T}\textsubscript{1} M\textsubscript{4}, where T\textsubscript{1} is the fundamental circuit matrix.

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

\[
R(ES) = \bigcup_{z \in R(HS)} R_(ES) = \bigcup_{z \in R(HS)} R_z(LS_1) p_{A_1} \times R_z(LS_2) p_{A_2}
\]

**Proof** R(ES) = \bigcup_{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 ∈ 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_1) p_{A_1} = R_z(LS_1) holds. Furthermore the general result

R(ES) p_{A_1} p_{A_2} \subseteq R(ES) p_{A_1} \times R(ES) p_{A_2}

holds for arbitrary disjoint sets of places P\textsubscript{1} and P\textsubscript{2}. Applying the results from Theorem 3 we get

R_z(ES) = R_z(ES) p_{A_1} p_{A_2} \subseteq 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}.

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\textsubscript{4} = (z, a\textsubscript{1}, a\textsubscript{2}) be such that z \in R(HS), (z, a\textsubscript{1}) \in R_z(LS_1) and (z, a\textsubscript{2}) \in R_z(LS_2) but M\textsubscript{4} \notin R_z(ES). According to Theorem 4 exists at least one fundamental circuit C with b\textsubscript{C}\cdot M^ES_0 \neq b\textsubscript{C}\cdot M\textsubscript{4}, where b\textsubscript{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\textsubscript{C}. Since M^LS_1 = M^ES_0 p_{N_1}, and R(LS\textsubscript{i}) = RS(ES) p_{L_{S\textsubscript{i}}}, C cannot be completely in LN\textsubscript{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\textsubscript{d} such that C\textsubscript{d} will be completely in LS\textsubscript{1} and the transformation will ensure that b\textsubscript{C\textsubscript{d}}\cdot M^ES_0 \neq b\textsubscript{C\textsubscript{d}}\cdot M\textsubscript{4} holds iff b\textsubscript{C}\cdot M^ES_0 \neq b\textsubscript{C}\cdot M\textsubscript{4}.

We partition C into two sets of paths P\textsubscript{1}, P\textsubscript{2} such P\textsubscript{1} contains the paths which are in LN\textsubscript{1} and P\textsubscript{2} contains the paths left in C. The paths in P\textsubscript{1}, P\textsubscript{2} are chosen to be of maximal length, such that C is the concatenation of paths taken alternately from P\textsubscript{1}, P\textsubscript{2}. Obviously any π \in P\textsubscript{2} is a path with a corresponding TT-MSIF x in EN (by definition of EN). Hence π together with x forms a fundamental circuit C\textsubscript{d} in EN. C\textsubscript{d} is completely in LN\textsubscript{2}, so by the same argumentation as above b\textsubscript{C\textsubscript{d}}\cdot M_0^ES = b\textsubscript{C\textsubscript{d}}\cdot M\textsubscript{4} must hold for the corresponding circuit vector b\textsubscript{C\textsubscript{d}} since (z, a\textsubscript{2}) is reachable in LS\textsubscript{2}. If we exchange path π in C by x we yield another fundamental circuit C\textsubscript{d}' which has a corresponding vector b\textsubscript{C\textsubscript{d}'} = b\textsubscript{C\textsubscript{d}} - b\textsubscript{x}. Now we get b\textsubscript{C\textsubscript{d}'}\cdot M_0^ES = b\textsubscript{C\textsubscript{d}'}\cdot M\textsubscript{4} = b\textsubscript{C\textsubscript{d}}\cdot M\textsubscript{4}+ b\textsubscript{x}\cdot M\textsubscript{4} = b\textsubscript{C}\cdot M\textsubscript{4}, which means that we can regard C\textsubscript{d} instead of C to argue that M\textsubscript{4} is not reachable. This argumentation can be applied stepwise and consecutively for every element of P\textsubscript{2}, such that finally all elements of P\textsubscript{2} are exchanged with their corresponding TT-MSIFs and the resulting circuit C\textsubscript{d} is completely in LN\textsubscript{1} and b\textsubscript{C\textsubscript{d}}\cdot M_0^ES \neq b\textsubscript{C\textsubscript{d}}\cdot M\textsubscript{4}. But this contradicts (z, a\textsubscript{1}) \in R_z(LS\textsubscript{1}).

The previous theorem allows the representation of R(ES) using R(HS) and R(LS\textsubscript{i}). The reachability set of the complete net can be completely characterized knowing the local reachability sets R(LS\textsubscript{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 (kroncker) products and sums [2, 5, 11] to combine submatrices related to low level systems. Let Q be the n × n generator matrix and let Q[z, z'] be the n(z) × n(z') submatrix describing transitions between markings from R_z(ES) and R_z'(ES). Q is block-structured in these submatrices and to every block Q[z, z'] belongs one element Q_{HS}(z, z'). The different blocks can be computed from the matrices of the
low level systems as follows.

\[
Q[z, z'] = \begin{cases} 
Q_1[t, z, z] \circ Q_2[t, z, z] & \text{if } z = z' \\
Q_1[t, z, z'] \circ U_2[t, z, z'] & \text{if } Q_{HS}(z, z') = t \\
U_1[t, z, z'] \circ Q_2[t, z, z'] & \text{if } Q_{HS}(z, z') = t \\
0 & \text{otherwise}
\end{cases}
\] (1)

The above representation can be used to generate \(Q\) 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 \(Q\), 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 straightforward (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.

1. Decompose the MG in \(LS_1, LS_2\) and \(HS\) (step 1)
2. Compute initial aggregates for \(LS_i\) (step 2)
3. REPEAT
   - Analyze \(LS_i\), where \(LS_i\) 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 corresponding aggregate or estimate the solution (step 5)
4. UNTIL convergence of the solution (step 6)
5. Estimate the approximation error (step 7)
6. IF approximation error too large THEN
   - Perform iterative solution (step 8)
7. 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 \(T_i^{\text{out}}\). So, every \(t \in T_i^{\text{out}}\) 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 \(LS_i\), 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 \(\lambda_t\) 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 \(T_i^{\text{in}}\), which represent an aggregate for \(LS_j\), are denoted as \(\mu_t(z)\), where \(t \in T_i^{\text{in}}\) 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 \(\mu_t\), 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.

\[
y_t[z]Q_t[r, z, z'] + \sum_{t \in T_i^{\text{out}}} Q_i[t, z, z']G_i[z', z] = 0 \quad (2)
\]
and \( y_i(z) e^T = 1.0 \), where \( z[t > z', e^T \) denotes the unit-
vector and \( G_i[z, z'] \) is a non-negative \( n_1(z) \times n_1(z') \)
matrix with unit row sum.

The choice of the G-matrices is in principle arbitrary, sometimes they might be realized by a corre-
spending U-matrices, if they exist, such that the equations describe the short-circuited low level sys-
tem. From the vector \( y_i(z) \) the aggregate parameters can be computed as
\[
\mu_i(z) = y_i(z) Q[t, z, z'] e^T.
\] (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 gen-
erator matrix \( Q' \) of the resulting system can be eas-
ily derived from equation (1). \( Q' \) is, like \( Q \), block-
structured corresponding to the the marking of the places \( IP \). Block \( Q'[z, z] \) is computed with the same
formulas than \( Q[z, z'] \) except all matrices belonging to \( LS_j \) are substituted by the corresponding aggre-
gate parameters. Thus, a non-zero matrix \( U_{i[j]}[t, z, z'] \)
is substituted by 1.0 and a matrix \( Q_{i[j]}[t, z, z'] \)
is sub-
tituted by \( \mu_i(z) \). Observe that the number of mark-
ings in \( R_i(ES) \) 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 \( Q^2 \) and \( Q^3 \),
when \( Q \) is represented exploiting the tensor structure.
Analysis of \( LS_i \) means the solution of the system
\[
p^i Q^i = 0 \text{ and } p^i e^T = 1.0.
\] (4)

Each vector \( p^i \) can be decomposed into subvectors
\( p^i[z] \) including probabilities for states from \( R_i(ES_i) \),
due to the ordering of states \( p^i = (p^i[1], \ldots, p^i[n_{HS}]) \).
Define \( \tilde{p}^i[z] \) as \( p^i[z]/(p^i[z] e^T) \), i.e. the subvector is
normalized to 1. New aggregate parameters, in step
4, are computed from \( p^i \) as
\[
\mu_i(z) = \tilde{p}^i[z] Q[t, z, z'] e^T, \quad (5)
\]
which yields an aggregate with marking dependent pa-
rameters, similar to the approach chosen in [7]. The
aggregate parameters for \( LS_j \) reflect the conditional
throughput of \( t \in T^n \) with marking \( z \) on the places
\( IP \). In [3, 8] aggregates with state independent pa-
rameters, approximating the mean delay of a low level
system have been introduced. Corresponding aggre-
gate parameters can be computed as follows.
\[
\mu_i(z) = \mu_t = \frac{\sum_{z \neq z'} p^i[z] Q[t, z, z'] e^T}{\sum_{z \neq z'} \| p^i[z] e^T \|}
\] (6)

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

Observe that (5) and (6) are not identical to the corre-
sponding 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 pro-
posed in the mentioned paper compute aggregate pa-
rameters in a more complex way using net-level re-
sults. However, this introduces additional effort and,
possibly, convergence problems. Nevertheless, other
forms of aggregate parameters can be integrated in our framework and do not affect the gen-
eral approach. At the current stage it is not possible
to decide in general which form of parameter computa-
tion gives the best results, although examples indi-
cate 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
iteration [9].

Step 5 is optional but needs not much effort, it is re-
quired if the overall solution should be approximated
by means of the solutions \( p^i \) and it might also be use-
ful 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 struc-
ture as \( Q_{HS} \). The matrix elements can be computed
from (1) after substituting all non-zero U-matrices by
1.0 and all matrices \( Q[t, z, z'] \) by \( \mu_i(z) \). We denote
the resulting matrix by \( Q^0 \), since it can be directly
derived from (1), no additional data structures are re-
quired for this matrix. The stationary solution \( p^0 \)
for this system is computed via (4) setting \( i = 0 \). From
\( p^0 \) the throughput of \( HS \) can be computed and com-
pared with the throughputs determined from \( p^i \)
and \( p^0 \), 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 ap-
proximate method converges and even if it does, the
quality of the gained approximate results is unknown.
However, experience shows that usually only a few it-
erations are required before convergence is reached.
Convergence is assumed if the parameters computed
for the aggregates or the solution vectors for the ag-
gregated systems differ only by a small factor in con-
secutive iterations. Let \( p^i(k) \) and \( p^i(k) \) 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
\[
\| p^i(k) - p^i(k-1) \| \leq \epsilon_1
\] (7)
\[
\| p^2(k)[z] e^T - p^2(k)[z] e^T \| \leq \epsilon_2
\] (8)
holds. Convergence according to the aggregate pa-
rameters is given when
\[
\frac{\mu_i(k)(z) - \mu_i(k-1)(z)}{\mu_i(k-1)(z)} \leq \epsilon \quad \text{for all } z \in [1, \ldots, n_{HS}]
\]
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

\[ \tilde{p}[z] = p^0(z) \tilde{p}^1[z] \odot \tilde{p}^2[z] \] (9)

Let \( p \) be the stationary solution of the complete system (i.e. \( pQ = 0 \) and \( pe^T = 1.0 \)). The error of the approximation is described by \( ||p - \tilde{p}|| \), however, since \( p \) is unknown it cannot be computed exactly. An alternative is to check the residuals \( ||\tilde{p}Q|| \) 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 \( Q \) by exploitation of the tensor structure, and they can even be computed without a memory expensive representation of \( \tilde{p} \) since elements of \( \tilde{p} \) 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 \( \tilde{p} \). 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 \( LS_1 \) or \( HS \). Thus they are computed directly from the vector \( p_i \) \( (i = 0, 1, 2) \) or, if \( \tilde{p} \) has been improved by an iterative technique, by mapping \( \tilde{p} \) on \( p_i \), 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 \( P_1, P_2 \) and \( P_3 \). We assume that all transitions have single server semantics and, apart from \( T_2, T_3, T_5 \) and \( T_7 \), have transition rate 1.0. Transitions \( T_2 \) and \( T_6 \) have rate \( \lambda \), transitions \( T_3 \) and \( T_7 \) rate \( \mu \).

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 LS1, b) low level system LS2, and c) high level system HS for example SMG](image-url)

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 \( Q \) \( (#nz(Q)) \), which is shown in the third column, grows even faster. For a numerical analysis using standard techniques and sparse storage of \( Q \) the required space is proportional to \( 2n_{ES} + #nz(Q) \). If the tensor structure is exploited in an iterative solution, then the required storage is proportional to \( 2n_{ES} + #nz(Q, U) \),
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 an 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 $P_1$. 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 $T_7$ and $T_7$. Aggregates with fixed transition rates do not 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 $p$ with the solution reached after $iter$ iteration steps. The initial distribution $e_1$ 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 a bad convergence. The uniform distribution denoted by $1/ne$ performs much better. The values in column $p_1$ 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 $1/ne$ as an approximation of $p$ yield much larger errors than for $p_1$. $p_2$ 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 $10^{-6}$, 160 iterations are needed starting with $e_1$, with $1/ne$ and $p_1$, 130 iterations are required and starting with $p_2$ 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.

### Table 1: Reachability set and matrix size of the example

<table>
<thead>
<tr>
<th>$K$</th>
<th>$n_{ES}$</th>
<th>$#nz(Q)$</th>
<th>$n_{HS}$</th>
<th>$n_{LS_1}$</th>
<th>$n_{LS_2}$</th>
<th>$#nz(Q_1, U_i)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>228</td>
<td>876</td>
<td>16</td>
<td>88</td>
<td>64</td>
<td>635</td>
</tr>
<tr>
<td>5</td>
<td>1704</td>
<td>8224</td>
<td>36</td>
<td>476</td>
<td>243</td>
<td>2995</td>
</tr>
<tr>
<td>10</td>
<td>38476</td>
<td>234850</td>
<td>121</td>
<td>5201</td>
<td>1633</td>
<td>36686</td>
</tr>
<tr>
<td>15</td>
<td>225348</td>
<td>1486488</td>
<td>255</td>
<td>22926</td>
<td>5173</td>
<td>155721</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>$iter$</th>
<th>$e_1$</th>
<th>$1/ne$</th>
<th>$p_1$</th>
<th>$p_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>9.93e-01</td>
<td>7.07e-03</td>
<td>2.54e-02</td>
<td>4.31e-04</td>
</tr>
<tr>
<td>10</td>
<td>7.10e-02</td>
<td>5.73e-03</td>
<td>5.97e-03</td>
<td>1.02e-04</td>
</tr>
<tr>
<td>50</td>
<td>2.36e-03</td>
<td>4.96e-04</td>
<td>4.63e-04</td>
<td>1.17e-06</td>
</tr>
<tr>
<td>100</td>
<td>1.02e-04</td>
<td>2.17e-05</td>
<td>2.03e-05</td>
<td>5.03e-08</td>
</tr>
<tr>
<td>300</td>
<td>3.56e-10</td>
<td>8.01e-11</td>
<td>7.48e-11</td>
<td>1.76e-13</td>
</tr>
</tbody>
</table>
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 Q with minimum storage requirements.

References


