A response time approximation technique for stochastic general P/T systems*

Carlos J. Pérez-Jiménez and Javier Campos
Departamento de Informática e Ingeniería de Sistemas
Centro Politécnico Superior, Universidad de Zaragoza
María de Luna 3, E-50015 Zaragoza, Spain
e-mail: cjperez, jcampos @posta.unizar.es.

ABSTRACT

Stochastic Petri nets is a well–known formalism adequate for the design, validation, and performance evaluation of discrete event and manufacturing systems. In this paper, we deal with steady–state throughput approximation of complex concurrent systems modelled with stochastic Petri nets. More precisely, we generalize to arbitrary stochastic P/T systems a response time approximation technique that was firstly proposed for special net subclasses. The presented technique is based on the divide and conquer principle and it is achieved in two steps. The first one, a net–driven decomposition of the model into several subsystems and the second one, an iterative solution algorithm that computes a throughput approximation of the original model transitions based on the solution of the embedded continuous time Markov chain of the subsystems. Experimental results on several examples generally have an error of less than 5%, and the state space is usually reduced by more than one order of magnitude.

1 INTRODUCTION

Stochastic Petri nets [7] is a well–known formalism adequate for the design, validation, and performance evaluation of discrete event and manufacturing systems. In this paper, we deal with *steady–state throughput approximation* of complex concurrent systems modelled with stochastic Petri nets. More precisely, we generalize to arbitrary stochastic P/T systems a *response time approximation* technique that was firstly proposed for *marked graphs* [1] and later extended to *weighted T–systems* [9] and *deterministic systems of sequential processes* [10, 8].

The presented technique, based on the *divide and conquer* principle, is achieved in two steps: (1) a net–driven decomposition of the model into several subsystems, that requires a reduction rule in order to be able to build abstract views of some modules of the original system (we use a reduction rule that has been proposed in [2] in the framework of *tensor algebra–based exact solution* of stochastic nets); and (2) an iterative solution algorithm that computes a throughput approximation of the original model transitions based on the solution of the embedded continuous time Markov chain (CTMC) of the subsystems. The gain of the technique with respect to the classical exact solution algorithm (derivation and solution of the embedded CTMC of the original model) is in both *memory and time requirements*. With respect to space, we never store the infinitesimal generator of the CTMC of the whole system. Instead of that, the generator matrices of smaller subsystems are stored (it must be pointed out that the dimension of those matrices is equal to the number of reachable states of the model, and that this number increases exponentially with the size of the net model). With respect to time complexity, we do not solve the CTMC isomorphous to the original system but those isomorphous to the derived subsystems. The price we must pay for the above savings is that only approximate values are computed for the throughput of transitions.

We assume that the reader is familiar with concepts and notation of P/T nets [5] and stochastic nets [7]. The paper is organized as follows. In section 2, we overview the decomposition technique for general Petri nets that has been proposed in [2]. For our purpose, that decomposition technique has a main problem: *the derived subsystems may be non-ergodic* (thus, the solution of the embedded CTMC makes no sense). In section 3, we solve the mentioned problem by presenting a very simple technique to build ergodic CTMC from the subsystems. Even though that technique allows to compute a (meaningful) solution in all cases, in some situations the result may be not very accurate due to the inclusion of *spurious* states in the subsystems that do not correspond to actual ones in the original models. For solving more accurately those cases, we present in section 4 a way of eliminating the spurious solutions, with an additional storage and time cost. In section 5, the iterative response time approximation algorithm (similar to that presented in [1]) is explained as well as an example of application to a non-trivial example. Some concluding remarks are stressed in section 6.

2 STRUCTURAL DECOMPOSITION OF PN SYSTEMS AND TWO-LEVEL ABSTRACT VIEWS

In this section, we overview a decomposition technique for general PN systems that has been proposed in [2] in the framework of *tensor algebra-based exact*

*This work has been developed within the project HCM CT94-0452 (MATCH) of the European Union.

solution of stochastic nets.

Structured view of PN systems

An arbitrary PN system can always be observed as a set of *modules* (disjoint simpler PN systems) that asynchronously communicate by means of a set of *buffers* (places). That structured view of the system is recalled in the next definition.

Definition 1 [2] *A strongly connected PN system, $\mathcal{S}=\langle P_1\cup\ldots\cup P_K\cup B, T_1\cup\ldots\cup T_K, \mathbf{Pre}, \mathbf{Post}, \mathbf{m_0}\rangle$, is a* System of Asynchronously Communicating Modules*, or simply a SAM, if:*

1. $P_i \cap P_j = \emptyset$, $\forall i,j \in \{1,\ldots,K\}, i \neq j$;
2. $T_i \cap T_j = \emptyset$, $\forall i,j \in \{1,\ldots,K\}, i \neq j$;
3. $P_i \cap B = \emptyset$, $\forall i \in \{1,\ldots,K\}$;
4. $T_i = P_i^\bullet \cup {}^\bullet P_i$, $\forall i \in \{1,\ldots,K\}$.

$\langle \mathcal{N}_i, \mathbf{m_0}_i\rangle = \langle P_i, T_i, \mathbf{Pre}_i, \mathbf{Post}_i, \mathbf{m_0}_i\rangle, i \in \{1,\ldots,K\}$, *are called* modules *of $\mathcal{S}$ (where $\mathbf{Pre}_i$,$\mathbf{Post}_i$, and $\mathbf{m_0}_i$ are the restrictions of $\mathbf{Pre}$, $\mathbf{Post}$, and $\mathbf{m_0}$ to P_i and T_i). The places B are called* buffers*. Transitions belonging to the set $\mathrm{TI} = {}^\bullet B \cup B^\bullet$ are called* interface *transitions. The remaining ones ($(T_1\cup\ldots\cup T_K)\setminus \mathrm{TI}$) are called* internal *transitions.*

We remark that all the strongly connected PN systems belong to the SAM class. The only addition introduced in the above definition is a *structured view* of the model (either given by construction or decided after observation of the model). Many structured views are possible, ranging from the extreme consideration of each transition as a different module (all the places being buffers) to consider that the system is a single module (and there are no buffers).
With respect to timing interpretation, we assume that independent, exponentially distributed random variables are associated to the firing of transitions with single server semantics as in classical *stochastic PN's* [7].
The final goal of our work being to approximate the steady-state throughput of transitions of the model, we assume that such (unique) steady-state behaviour exists. Even more, we restrict to *structurally live, structurally bounded* (therefore *consistent* and *conservative*) and *reversible* (therefore *ergodic*) PN systems.

Reduction rule and abstract views

In [2], the reduction rule that follows has been introduced for the *internal behaviour* of modules of a SAM. Informally speaking, each module is decomposed into several pieces and each piece is substituted by a set of new places. Later, using that reduction, the original model can be decomposed into a collection of *low level systems* and a *basic skeleton*. In each low level system, only one module is kept while the internal behaviour of the others is reduced. In [2], the low level systems and the basic skeleton are used for a tensor algebra-based exact computation of the embedded CTMC. In this paper, we use the decomposition for a non-exact but more efficient approximation of the throughput of transitions.

Definition 2 *Let $\mathcal{S} = \langle P, T, \mathbf{Pre}, \mathbf{Post}, \mathbf{m_0}\rangle$ be a SAM with $P = P_1\cup\ldots\cup P_K\cup B$ and $T = T_1\cup\ldots\cup T_K$. The equivalence relation R is defined on $P \setminus B$ by: $\langle p, p'\rangle \in \mathrm{R}$ for $p, p' \in P_i$ iff there exists a non-directed path Π in $\mathcal{N}_i$ from p to p' such that $\Pi \cap \mathrm{TI} = \emptyset$ (i.e., containing only internal transitions). The different equivalence classes defined in P_i, $i = 1,\ldots,K$, by the relation R are denoted as P_i^j, $j = 1,\ldots,r(i)$.*

The next step in the reduction process is to add to the modules $\mathcal{N}_i$ a set H_i^j of *marking structurally implicit places* for each equivalence class P_i^j, $j = 1,\ldots,r(i)$, defined by R in $\mathcal{N}_i$.

Definition 3 [4] *Let $\mathcal{N}$ be a net and p be a place with incidence vector $l_p = \mathbf{C}[p,\cdot]$. The place p is a* marking structurally implicit place *(MSIP) in $\mathcal{N}$ if there exists $\mathbf{y} \geq \mathbf{0}$ such that $\mathbf{y}[p] = 0$ and $l_p = \mathbf{y}\cdot\mathbf{C}$. The set of places in $\|\mathbf{y}\|$ are called* implying places *of p (where $\|\mathbf{y}\|$, called support of $\mathbf{y}$, is the set of non-zero components of $\mathbf{y}$).*

In [4], an efficient method for computing an initial marking for a MSIP for making the place *implicit* is also presented (a place is implicit, under interleaving semantics, if it can be deleted without changing the firing sequences).
An algorithm for the computation of a set H_i^j of MSIP's for each equivalence class P_i^j defined in a module has been proposed in [2]. The basic idea is to consider all the MSIP's $p_\mathbf{y}$ derived from the *minimal P-semiflows* $\mathbf{y}$ of the subnet induced by P_i^j (i.e., $p_\mathbf{y}$ is the place with incidence vector $l_{p_\mathbf{y}} = \mathbf{y}\cdot\mathbf{C}$, where $\mathbf{y}$ is such that $\mathbf{y}\cdot\mathbf{C}[P_i^j, T_i^j] = 0, \mathbf{y} \geq 0$, $\mathbf{y}$ has minimal support).
The next step for the definition of the low level systems and the basic skeleton is to define an *extended system $\mathcal{ES}$*.

Definition 4 [2] *Let $\mathcal{S} = \langle P, T, \mathbf{Pre}, \mathbf{Post}, \mathbf{m_0}\rangle$ be a SAM with $P = P_1\cup\ldots\cup P_K\cup B$ and $T = T_1\cup\ldots\cup T_K$. The extended system $\mathcal{ES}$ is obtained from $\mathcal{S}$ by adding all the places in H_i^j, $j = 1,\ldots,r(i), i = 1,\ldots,K$, with the initial marking necessary for making them implicit.*

Consider, for instance, the SAM system given in Fig. 1.a. It is composed of three modules, interconnected through 5 buffers. Places b1 to b5 are the buffers, while places whose tag starts with a, c, d identify the first, second and third component, respectively. Places that start with IP are the implicit places computed with the previous algorithm. Due to the strong interconnections between the components and the buffer places, and to the high number of interface transitions (12), the algorithm produces 14 implicit

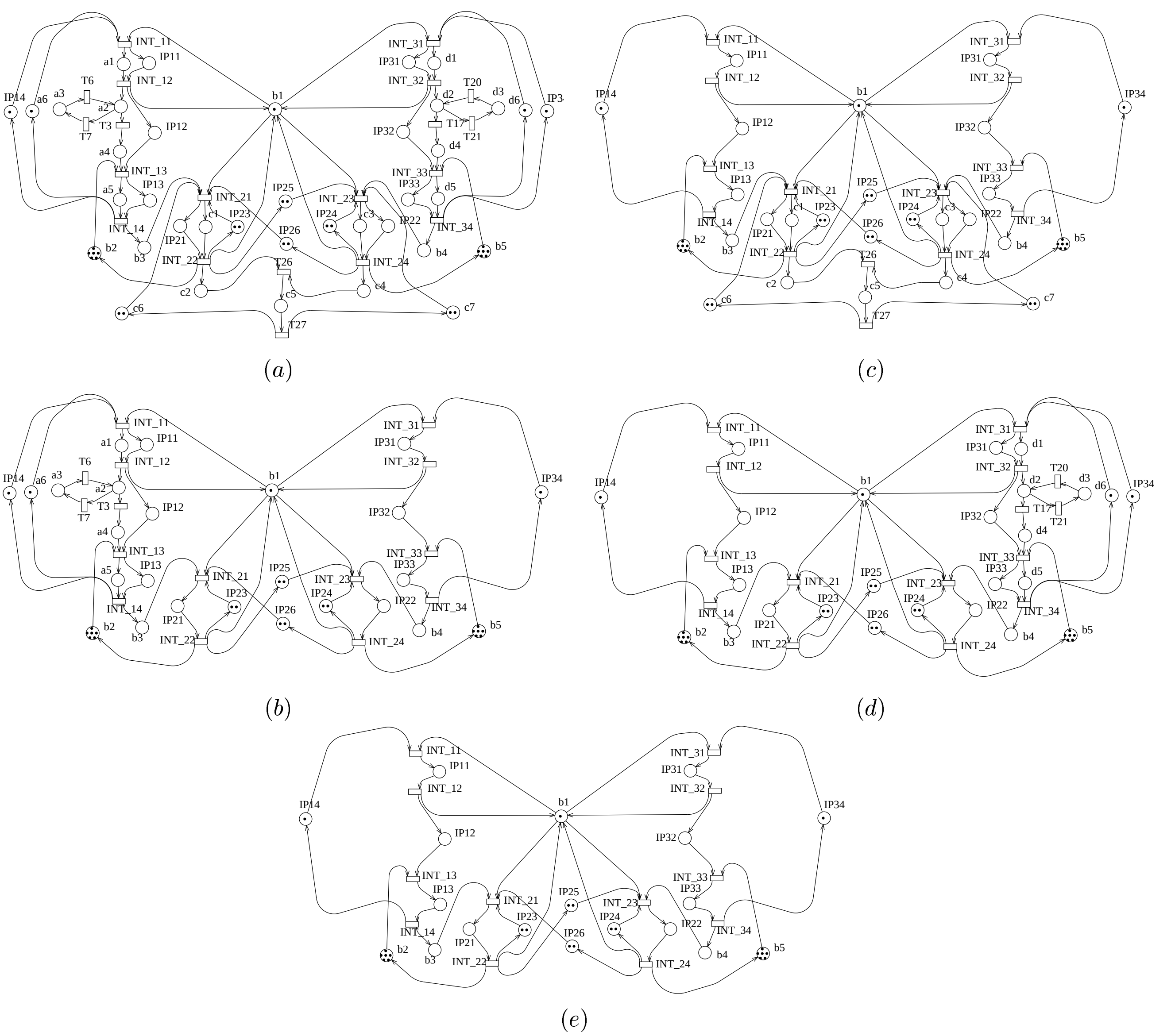

Figure 1: (a) A system, (b) its $\mathcal{LS}_1$, (c) $\mathcal{LS}_2$, (d) $\mathcal{LS}_3$ and (e) $\mathcal{BS}$

places, moreover 8 of these places are exact replicas of places of the net, therefore we should not expect to have very "abstract" macro states, and, consequently, the advantage of the structured approach in this case is going to be rather limited.
From the extended system, K different *low level systems* $\mathcal{LS}_i$ can be obtained deleting all places in P_j, $j \neq i$, and transitions in $T_j \setminus \mathrm{TI}$, $j \neq i$ ($i = 1, \ldots, K$).

Definition 5 [2] *Let* $\mathcal{S} = \langle P_1 \cup \ldots \cup P_K \cup B, T_1 \cup \ldots \cup T_K, \mathbf{Pre}, \mathbf{Post}, \mathbf{m_0} \rangle$ *be a SAM and* $\mathcal{ES}$ *its corresponding extended system.*

i) The low level system $\mathcal{LS}_i$ $(i = 1, \ldots, K)$ *of* $\mathcal{S}$ *is the system obtained from* $\mathcal{ES}$ *deleting all the nodes in* $\bigcup_{j \neq i}(P_j \cup (T_j \setminus \mathrm{TI}))$ *and their adjacent arcs.*

ii) The basic skeleton $\mathcal{BS}$ *of* $\mathcal{S}$ *is the system obtained from* $\mathcal{ES}$ *deleting all the nodes in* $\bigcup_j (P_j \cup (T_j \setminus \mathrm{TI}))$ *and their adjacent arcs.*

In other words, in each $\mathcal{LS}_i$ all the modules $\mathcal{N}_j$, $j \neq i$, are reduced to their interface transitions and to the implicit places that were added in the extended system, while $\mathcal{N}_i$ is fully preserved. Systems $\mathcal{LS}_i$ represent different low level views of the original model. In the $\mathcal{BS}$ all the modules are reduced in the same way, and it constitutes a high level view of the system.
By construction, since the original net is conservative then the $\mathcal{LS}_i$ and the $\mathcal{BS}$ are also conservative, so the reachability sets of all these subsystems are finite.
For the SAM in Fig. 1.a, the low level systems $\mathcal{LS}_1$, $\mathcal{LS}_2$, $\mathcal{LS}_3$, and the $\mathcal{BS}$ are depicted in Fig. 1.b, 1.c, 1.d

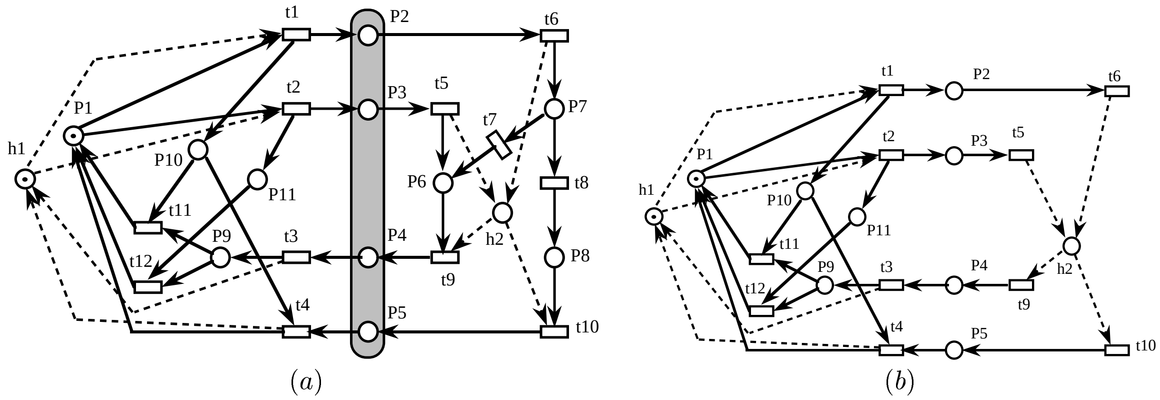

Figure 2: (a) A system and (b) its low level non live system

and 1.e, respectively.

Property 6 [2] *Let $\mathcal{S}$ be a SAM, $\mathcal{LS}_i$ its low level systems (i=1,..., K), $\mathcal{BS}$ its basic skeleton, and* $\mathrm{L}(\mathcal{S})$ *the language of firing sequences of $\mathcal{S}$. Then*

i) $\mathrm{L}(\mathcal{S})\mid_{T_i\cup \mathrm{TI}}\subseteq \mathrm{L}(\mathcal{LS}_i)$, *for* $i=1,\ldots,K$.

ii) $\mathrm{L}(\mathcal{S})\mid_{\mathrm{TI}}\subseteq \mathrm{L}(\mathcal{BS})$.

The above property states that the reduction technique presented here does not remove but eventually adds new paths between interface transitions.

3 GUARANTEEING SUBSYSTEMS ERGODICITY

From the result stated in Property 6, it follows that the reachability graphs (RG's) of $\mathcal{LS}_i$ and $\mathcal{BS}$ include at least the projections on the corresponding preserved nodes of the reachable markings of the original system. They also reproduce (at least) the projections on the preserved transitions of the firing sequences of the original system. But since the inclusion in the statement of Prop. 6 is not strict, the RG's of the subsystems may eventually include new (let us say) *spurious* markings and firing sequences that do not correspond to actual markings and firing sequences of the original system. In some cases, this non desired behaviour can lead to non ergodic systems. In this section we present a technique to avoid such undesired behaviour, guaranteeing ergodicity of the subsystems.
Consider, for example, the system given in Fig. 2.a. Cutting the system through the places P_2 to P_5 and applying the reduction technique of the previous section to the right hand side subnet, the $\mathcal{LS}_1$ of Fig. 2.b is obtained. In the original system after the firing of interface transition t_5 only transition t_9 can be fired, but in $\mathcal{LS}_1$ it is also possible to fire transition t_{10} after the firing of transition t_5. This new possible firing makes $\mathcal{LS}_1$ non live (the sequence $\sigma = t_2t_5t_{10}$ is firable in $\mathcal{LS}_1$ and the marking produced by the firing of σ is P_5P_{11} that is a total deadlock).
In other words, the embedded CTMC of the subsystems may be non ergodic. These CTMC must be adjusted in order to make them available for subsequent computations.
There is a direct way to achieve this objective. In general, the RG's of the subsystems may have several strongly connected components. To obtain an ergodic CTMC only the strongly connected component of the initial marking in each subsystem must be selected. It will be proved that these strongly connected components include, at least, all the projected states and firing sequences of the original net system.

Theorem 7 *Let $\mathcal{S}$ be a SAM, $\mathcal{ES}$, $\mathcal{LS}_i (i=1,\ldots,K)$ and $\mathcal{BS}$ its extended, low level and the basic skeleton systems, respectively. Let* $\mathrm{RG}^*(\mathcal{LS}_i)$ *and* $\mathrm{RG}^*(BS)$ *be the strongly connected components of* $\mathrm{RG}(\mathcal{LS}_i)$ *and* $\mathrm{RG}(BS)$, *respectively, that contain the initial marking. Let* $\mathrm{L}^*(\mathcal{LS}_i)$ *and* $\mathrm{L}^*(BS)$ *be the language of firing sequences of these ergodic reachability subgraphs of $\mathcal{LS}_i$ and $\mathcal{BS}$, respectively. Then*

i) $\mathrm{L}(\mathcal{S})\mid_{T_i\cup \mathrm{TI}}\subseteq \mathrm{L}^*(\mathcal{LS}_i), (i=1,\ldots,K)$,
$\mathrm{L}(\mathcal{S})\mid_{\mathrm{TI}}\subseteq \mathrm{L}^*(\mathcal{BS})$.

ii) $\mathrm{RS}(\mathcal{ES})\mid_{P_i\cup H\cup B}\subseteq \mathrm{RS}^*(\mathcal{LS}_i), (i=1,\ldots,K)$,
$\mathrm{RS}(\mathcal{ES})\mid_{B\cup H}\subseteq \mathrm{RS}^*(\mathcal{BS})$.

Proof:
Consider the extended system $\mathcal{ES}$ of $\mathcal{S}$. By definition, all the places in $H=\bigcup_{i=1}^{K} H_i$ are implicit in $\mathcal{ES}$. Therefore $\mathrm{L}(\mathcal{S}) = \mathrm{L}(\mathcal{ES})$. Since $\mathcal{S}$ is reversible, $\mathbf{m_0}$ is a *home state* (in the case that $\mathcal{S}$ is not reversible but it still has home state, any home state reachable from $\mathbf{m_0}$ can be considered as the new initial marking). Given a marking $\mathbf{m}\in \mathrm{RS}(\mathcal{ES})$, by reversibility of $\mathcal{ES}$, there exists $\sigma,\tau\in \mathrm{L}(\mathcal{ES})$ such that $\mathbf{m_0}\xrightarrow{\sigma}\mathbf{m}\xrightarrow{\tau}\mathbf{m_0}$. Let $\mathbf{m_0}_i = \mathbf{m_0}\mid_{P_i\cup H\cup B}$ be the initial marking of $\mathcal{LS}_i$ and $\mathbf{m}' = \mathbf{m}\mid_{P_i\cup H\cup B}$ be the projection of $\mathbf{m}$ over the places of $\mathcal{LS}_i$. Let $\sigma' = \sigma\mid_{T_i\cup\mathrm{TI}}, \tau' = \tau\mid_{T_i\cup \mathrm{TI}}$. It must be proved that $\mathbf{m}'\in \mathrm{RS}^*(\mathcal{LS}_i)$ and $\sigma'\in \mathrm{L}^*(\mathcal{LS}_i)$. By definition of $\mathcal{ES}$, it is clear that the marking of the places in $H\cup B$ is only changed by the firing of interface transitions, and the marking

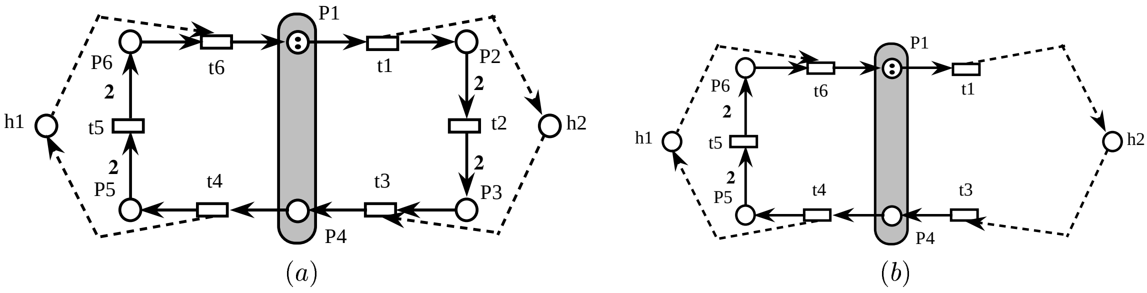

Figure 3: (a) A system and (b) its low level system

of the places in P_i is only changed by the firing of transitions in T_i. Then, in $\mathcal{LS}_i$, $\mathbf{m_0}_i \xrightarrow{\sigma'} \mathbf{m}' \xrightarrow{\tau'} \mathbf{m_0}_i$, so $\mathbf{m}' \in \mathrm{RS}^*(\mathcal{LS}_i), \sigma' \in \mathrm{L}^*(\mathcal{LS}_i)$. $\diamondsuit$

The strongly connected components of a directed graph can be efficiently computed with a time complexity of $O(max(n, \mid E \mid))$, where n is the number of nodes of the graph and $\mid E \mid$ the number of edges [6].

The above theorem gives a general technique applicable to any structurally live, structurally bounded and reversible SAM to obtain, from the subsystems, ergodic CTMC available for subsequent computations.

In the case of Fig. 2, $\mathrm{RG}(\mathcal{LS}_1)$ has two strongly connected components. One of them with all the states but P_5P_{11} and the other one with only the state P_5P_{11} (the deadlock marking). The strongly connected component of the initial marking is the first one and then, the deadlock marking P_5P_{11} has been removed in the process.

It must be pointed out that the $\mathrm{RG}^*(\mathcal{LS}_i)$ and $\mathrm{RG}^*(\mathcal{BS})$ computed before (even leading to ergodic CTMC) may still include spurious markings (and/or firing sequences) that do not correspond to the projection of any marking (firing sequence) of the original system over the preserved nodes (transitions). In other words, the set of projections of the reachable markings (firing sequences) of the original system over the places (transitions) of the subsystem may be a proper subset of the reduced reachability set (language of firing sequences) of the corresponding subsystem.

For example, in Fig. 3.a a *weighted T-system* (the weighted extension of well-known *marked graphs*) is depicted. Cutting the system through the places P_1 and P_4 and applying the reduction technique of the previous section, the $\mathcal{LS}_1$ of Fig. 3.b is obtained. Now, the underlying CTMC of $\mathcal{LS}_1$ is ergodic, so computing in it the strongly connected component of the initial marking, the entire Markov chain is obtained. But in this Markov chain, there are spurious markings and firing sequences that were not possible in the original system. For example, in the original system, $\mathbf{m}[P_1] \cdot \mathbf{m}[P_4] = 0$, for any reachable marking, but in $\mathcal{LS}_1$, the marking P_1P_4 is reachable. This marking is not a projection of any reachable marking of the original system over the places of $\mathcal{LS}_1$.

4 REDUCING SPURIOUS MARKINGS IN THE SUBSYSTEMS

In many cases the technique developed in the previous section to guarantee the ergodicity of the subsystems is enough for getting good throughput approximations with the iterative algorithm presented in section 5, but in some cases, depending on the number of the spurious markings, these approximations may be very poor. In this section, a more complex technique is developed to reduce all the spurious states and some firing sequences, in order to improve the accuracy of the approximation in all the SAM systems.

To eliminate these spurious states from the strongly connected component of the reachability graph of the subsystems computed in the previous section, the structured construction of the reachability set of the original system developed in [2] will be used. In that paper, the same structural decomposition of a SAM system is used to obtain a tensor algebra expression of $\mathbf{G}$, the infinitesimal generator matrix of a superset of the reachability set of the original system in terms of the infinitesimal generator matrices of the subsystems. A brief explanation of the tensor algebra expression will be recalled here.

Given a SAM system, their $\mathcal{LS}_i$ and $\mathcal{BS}$ are derived. The reachable states of $\mathcal{LS}_i$ are classified according to their high level views on $\mathcal{BS}$ (their projection over the places of $\mathcal{BS}$, that will be denoted as $\mathbf{z}$). Then, a superset of $\mathrm{RS}(\mathcal{S})$ can be constructed from the reachability sets of the $\mathcal{LS}_i$. This superset of states has all the reachable states of $\mathcal{S}$ plus some spurious states that are not reachable in $\mathcal{S}$. The next step is to construct the infinitesimal generator $\mathbf{G}$ of this superset of states. To do that the following matrices are defined. If $\mathbf{Q}_i$ is the infinitesimal generator of $\mathcal{LS}_i$, we denote by $\mathbf{R}_i(\mathbf{z}, \mathbf{z}')$ the submatrix of $\mathbf{Q}_i$ with the transition rates from states of high level view $\mathbf{z}$ to states of high level view $\mathbf{z}$' in $\mathcal{LS}_i$.

For each $t \in \mathrm{TI}$ such that $\mathbf{z} \xrightarrow{t} \mathbf{z}'$ (the set of such transitions will be denoted as $\mathrm{TI}_{\mathbf{z},\mathbf{z}'}$) we define:

$$\mathbf{K}_i(t)(\mathbf{z}, \mathbf{z}')[\mathbf{m}, \mathbf{m}'] = \begin{cases} 1 & \text{if } \mathbf{m} \xrightarrow{t} \mathbf{m}', t \in \mathrm{TI} \\ 0 & \text{otherwise} \end{cases}$$

The matrix $\mathbf{K}_i(t)(\mathbf{z}, \mathbf{z}')$ is the adjacency submatrix of $\mathrm{RG}(\mathcal{LS}_i)$ with all the transitions from states of high

level view $\mathbf{z}$ to states of high level view $\mathbf{z}$' due to the firing of t.
From these matrices a tensor algebra expression of the infinitesimal generator $\mathbf{G}$ of the superset of states can be constructed [2]:

$$\begin{array}{rcl} \mathbf{G}(\mathbf{z},\mathbf{z}) & = & \bigoplus_{i=1}^{K} \mathbf{R}_i(\mathbf{z},\mathbf{z}) \\ \mathbf{G}(\mathbf{z},\mathbf{z}') & = & \sum_{t\in \mathrm{TI}_{\mathbf{z},\mathbf{z}'}} w(t) \bigotimes_{i=1}^{K} \mathbf{K}_i(\mathbf{z},\mathbf{z}') \end{array} \quad (1)$$

Notice that it is not necessary to store the entire supermatrix $\mathbf{G}$. The tensor algebra expression allow us to store only the small matrices $\mathbf{R}_i$ and $\mathbf{K}_i$.
The main properties of $\mathbf{G}$ are proved in the next theorem.

Theorem 8 [2] *Let $\mathcal{S}$ be a SAM system, $\mathcal{LS}_i$ its low level systems ($i = 1,\ldots,K$), and $\mathcal{BS}$ its basic skeleton. Let $\mathbf{G}$ be the matrix defined by equations (1). Then:*

1. $\forall$ $\mathbf{z}$ *and* $\mathbf{z}' \in \mathrm{RS}(\mathcal{BS})$*:* $\mathbf{R}(\mathbf{z},\mathbf{z}')$ *is a submatrix of* $\mathbf{G}(\mathbf{z},\mathbf{z}')$.
2. $\mathbf{m} \in \mathrm{RS}(\mathcal{S}), \mathbf{m}' \notin \mathrm{RS}(\mathcal{S}) \Rightarrow \mathbf{G}[\mathbf{m},\mathbf{m}'] = 0$.

As a consequence of the above theorem, by means of a *depth–first search* in $\mathbf{G}$ beginning at the initial marking we can travel through all the reachable markings of $\mathcal{S}$. By theorem 8, in this search spurious markings are never reached.
For an efficient implementation of the depth–first search in a graph, the adjacency list of nodes of a given vertex of the graph must be computed. In our case, using the tensor algebra expression of the matrix $\mathbf{G}$, it is possible to compute the states directly reachable from another. The algorithm to compute the adjacency list of a given state is the following:

Algorithm 4.1
input: *a marking* $\mathbf{m}$ *of the superset of* $\mathrm{RS}(\mathcal{S})$
$Ad := \emptyset$
for $i := 1$ **to** K **do**
  $\mathbf{m}_i := \mathbf{m}\,|_{P_i\cup H\cup B}$
end for
for $i := 1$ **to** K **do**
  for $j := 1$ **to** $n_{\mathbf{z},i}$ **do**
    if $\mathbf{R}_i(\mathbf{z},\mathbf{z})(\mathbf{m}_i,\mathbf{m}_j) = 1$ **then**
      $Ad := Ad \cup (\mathbf{m}_1,\ldots,\mathbf{m}_{i-1},\mathbf{m}'_j,\mathbf{m}_{i+1},\ldots,\mathbf{m}_K)$
    end if
  end for
end for
for $t \in \mathrm{TI}_{\mathbf{z},\mathbf{z}'}$ **do**
  $i := 0$
  while t *firable in* $\langle \mathcal{LS}_i, \mathbf{m}_i\rangle$ **do**
    $i := i + 1$
    $\mathbf{m}'_i := \mathbf{m}_i[t\rangle$
  end while
  if $i = K$ **then**
    $Ad := Ad \cup (\mathbf{m}'_1,\ldots,\mathbf{m}'_K)$
  end if
end for
output: *adjacency list* Ad *of* $\mathbf{m}$

In the above algorithm, $n_{\mathbf{z},i}$ is the number of states in $\mathcal{LS}_i$ whose high level view is $\mathbf{z}$. Given a marking $\mathbf{m}$ of the superset of states, the low level views $\mathbf{m}_i$ of $\mathbf{m}$ are computed. Then, by equations (1), $\mathbf{G}(\mathbf{z},\mathbf{z})(\mathbf{m},\mathbf{m}') \neq 0$ iff $\mathbf{m}$ and $\mathbf{m}$' differ only in one low level view (computed in the two nested loops), and $\mathbf{G}(\mathbf{z},\mathbf{z}')(\mathbf{m},\mathbf{m}') \neq 0$ iff $\mathbf{m}_i \xrightarrow{t} \mathbf{m}'_i$ in each $\mathcal{LS}_i$ (computed in the last loop).
The next step is to derive the reduced infinitesimal generators $\mathbf{Q}^*_i$ of $\mathcal{LS}_i$, using the depth–first search of $\mathrm{RG}(\mathcal{S})$ as given in [6]. It can be implemented inside the searching because in this kind of searching each node and edge of the graph is visited only once. In the next algorithm the generation of Qi^* is presented.

Algorithm 4.2
for *each edge* $\mathbf{m} \xrightarrow{t} \mathbf{m}'$ *visited in* dfs **do**
  if $\mathbf{m}'$ *is new* **then**
    for $i := 1$ **to** K **do**
      $\mathbf{m}'_i := \mathbf{m}'\,|_{P_i\cup H\cup B}$
      add $\mathbf{m}'_i$ *to* $\mathbf{Q}^*_i$ *if it is new*
    end for
  end if
  if $t \in \mathrm{TI}$ **then**
    $\mathbf{z} := \mathbf{m}\,|_{B\cup H}$
    $\mathbf{z}' := \mathbf{m}'\,|_{B\cup H}$
    for $i := 1$ **to** K **do**
      $\mathbf{Q}^*_i(\mathbf{m}_i,\mathbf{m}'_i) := \mathbf{R}_i(\mathbf{z},\mathbf{z}')(\mathbf{m}_i,\mathbf{m}'_i)$
    end for
  else $\{t \in T_j$ *for only one* $j\}$
    $\mathbf{Q}^*_j(\mathbf{m}_j,\mathbf{m}'_j) := \mathbf{R}_i(\mathbf{z},\mathbf{z}')(\mathbf{m}_j,\mathbf{m}'_j)$
  end if
end for
output: *Reduced infinitesimal generators* $\mathbf{Q}^*_i$

In the above algorithm, dfs denotes the depth–first search method mentioned before. To construct the reduced CTMC of $\mathcal{LS}_i$, when we visit a new edge in $\mathrm{RG}(\mathcal{S})$, first of all we must test if the reached marking is new. In that case we add its projections to the corresponding $\mathbf{Q}^*_i$ (if these projections are new in $\mathbf{Q}^*_i$). If the fired transition t is an interface transition, then a new rate must be added to all the $\mathbf{Q}^*_i$ because the interface transitions are present in all $\mathcal{LS}_i$. If t is internal, then t is only visible in one $\mathcal{LS}_j$, so there is a change of state only in that $\mathcal{LS}_j$.
By construction, all the states of the reduced CTMC are projections of reachable states in $\mathcal{S}$. Therefore, all the spurious states have been deleted and the inclusion of theorem 7.i) becomes an equality now. With regard to the firing sequences, with this improved technique we can eliminate some spurious firing sequences but not all of them, so the inclusion of theorem 7.ii) is still true (we can not achieve the equality).

5 APPLICATION TO ITERATIVE THROUGHPUT APPROXIMATION METHODS

The technique for an approximate computation of the throughput is, basically, a *response time approximation method* [1, 10, 8, 9]. In each $\mathcal{LS}_i$ there is a unique

module of $\mathcal{S}$ with all its nodes and transitions. Then, in the $\mathcal{LS}_i$ the interface transitions of the module $\mathcal{N}_j$ (for all $j \neq i$) approximate the response time of the reduced part of module $\mathcal{N}_j$.
The algorithm is the following:

Algorithm 5.1
select *the modules*
derive $\mathcal{LS}_i$, $i = 1, \ldots, K$ *and* $\mathcal{BS}$
give *an initial service rate* $\boldsymbol{\mu}_i^{(0)}$ *for* $i = 2, \ldots, K$
$j := 0$ {*counter for iteration steps*}
repeat
  $j := j + 1$
  for $i := 1$ **to** K **do**
    solve $\mathcal{LS}_i$ *with*
      input: $\boldsymbol{\mu}_l^{(j)}$ *for each* $l < i$
        $\boldsymbol{\mu}_l^{(j-1)}$ *for each* $i < l \leq K$
      output: initial rates $\boldsymbol{\mu}_i$
        throughputs $\boldsymbol{\chi}_i^{(j)}$ *of* $\mathrm{TI} \cap T_i$
    solve *the* $\mathcal{BS}$ *with*
      input: $\boldsymbol{\mu}_l^{(j)}$ *for each* $l < i$
        $\boldsymbol{\mu}_l^{(j-1)}$ *for each* $i < l \leq K$
        $\boldsymbol{\mu}_i$ *and* $\boldsymbol{\chi}_i^{(j)}$ *of* $\mathrm{TI} \cap T_i$
      output: actual rates $\boldsymbol{\mu}_i^{(j)}$ $\mathrm{TI} \cap T_i$
  end for
until *convergence of* $\boldsymbol{\chi}_1^{(j)}, \ldots, \boldsymbol{\chi}_K^{(j)}$

In the above procedure, once the K modules have been selected and given some initial values $\boldsymbol{\mu}_i^{(0)}$ of the rates of the interface transitions of $\mathcal{N}_i$, $(i = 2, \ldots, K)$, the underlying CTMC of $\mathcal{LS}_1$ is solved. The selection of the initial values of interface transitions rates does not affect (under our experience) to the accuracy of the method. A simple option is putting the initial rate of the transitions in the original model. From the solution of that CTMC, the first estimation $\boldsymbol{\chi}_1^{(1)}$ of the throughput of the interface transitions of $\mathcal{N}_1$ can be computed. Then, the initial estimated values of service rates of interface transitions $\mathrm{TI} \cap T_1$ must be derived. To do that, we take the initial values $\boldsymbol{\mu}_1^{(0)}$ for service rates of transitions in $\mathrm{TI} \cap T_1$ and we search in the $\mathcal{BS}$ these rates such that the throughput of transitions in $\mathrm{TI} \cap T_1$ in the $\mathcal{BS}$ and $\boldsymbol{\chi}_1^{(1)}$ are equal. The same procedure is executed for each $\mathcal{LS}_i$ in a cyclic way. Each time we solve $\mathcal{LS}_i$ we obtain in the $\mathcal{BS}$ a new estimation of the interface transitions rates of $\mathrm{TI} \cap T_i$. The previous steps are repeated until *convergence* of the throughput approximations of the subsystems is achieved (if there exists).
With regard to the computation of the initial estimated values of the interface transitions rates of the $\mathcal{LS}_i$, we use the implicit places H_i in $\mathcal{LS}_i$ to compute the probability p_t that the transition $t \in \mathrm{TI} \cap T_i$ is enabled in the reduced subnet (similar to [1]). Then we put $\mu_t^{(j)} = \chi_t / p_t$.
The computation of the actual rates of the corresponding interface transitions in $\mathcal{BS}$ can be implemented with a multidimensional search. Now the net system ($\mathcal{BS}$) has considerably fewer states than the original one. In each iteration of this search, the underlying CTMC of the $\mathcal{BS}$ is solved. Note that only in the first iteration the CTMC is completely derived. For later iterations only some values must be changed. In some cases (for instance, for FRT-nets [3]), the relative throughputs (or visit ratios) of transitions of the $\mathcal{BS}$ are independent on the transition service time (they only depend on the net structure and on the conflict resolution rates). In these cases, only one parameter must be tuned up in the $\mathcal{BS}$, thus a unidimensional search can be implemented.
Now, convergence of the entire method and the uniqueness of the solution should be addressed. Although no formal proof gives positive answers so far to the above questions, extensive testing allows to conjecture that there exists one and only one solution, computable in a finite number of steps, typically between 2 and 6 if the convergence criterion is that the difference between the two last estimations of the throughput is less than 0.1%. With regard to the accuracy of the results, the extensive battery of numerical experiments has shown us that the error is less than 5% in all tested cases.
Now we are going to apply our approximation technique to the performance evaluation of the system of Fig. 1.a (taken from [2]). It is composed of three modules, interconnected through 5 buffers. The underlying CTMC of the original net has 38624 states, while the $\mathcal{LS}_i$ and the $\mathcal{BS}$ have 9922, 14960, 9922 and 6120 states, respectively. In this case, the large proportion of interface transitions with respect to the total number of transitions makes the $\mathcal{BS}$ state space of only one order of magnitude less than that of the original system.
The rates of transitions have been arbitrarily fixed as: 1.0 for transitions t_6, t_7,t_{20}, t_{21}, INT_{22}, INT_{24}; 2.0 for t_3, t_{17}, INT_{11}, INT_{21}, INT_{23}, INT_{31}; 3.0 for INT_{12}, INT_{14}, INT_{32}, INT_{34}; 4.0 for t_{26}, t_{27}; and 5.0 for INT_{13}, INT_{33}.
In this case, the exact throughput of transition INT_{11} is 0.249615. This is a system with visit ratios fixed by the net structure, so the search in the $\mathcal{BS}$ can be unidimensional (only one variable must be computed).
In Table 1 we present the iteration steps of the method for this case. Columns $\chi(T_i)$ are the estimated values for throughput of transition T_i at each iteration step. Columns 'scale f.' are the scale factors modifying the previous estimated service rates, computed with the $\mathcal{BS}$. Convergence of the method is usually obtained from the four iteration step. The error for this example was 0.0005%.
We have not proved formally the accuracy of the method, but in all tested cases the error of the method was below 5%. To initiate the iteration, our experience is that the method seems to be robust with respect to the initial values of the interface transitions.

6 CONCLUSIONS

An approximation technique was presented for approximate throughput computation of stochastic general Petri nets. The technique has two phases. A

$\mathcal{LS}_1$		$\mathcal{LS}_2$		$\mathcal{LS}_3$	
$\chi(\mathrm{INT}_{11})$	scale f.	$\chi(\mathrm{INT}_{21})$	scale f.	$\chi(\mathrm{INT}_{31})$	scale f.
0.0.250981	0.992185	0.252967	1.008583	0.250047	0.977260
0.0.250129	0.979808	0.249771	1.004615	0.249732	0.978686
0.0.249733	0.978763	0.249732	1.004615	0.249734	0.978744
0.0.249734	0.978752	0.249731	1.004606	0.249733	0.978744
Error: 0.0005%					

Table 1: Iteration results for the SAM in Fig. 1.a

first structural phase in which the original net is decomposed in several smaller subsystems, and a second quantitative phase in which an iterative response time approximation algorithm is used to compute a throughput approximation of the original system transitions. The gain of the technique with respect to the classical exact solution algorithm (solution of the embedded CTMC of the original model) is in both memory and time requirements. With respect to space, the infinitesimal generator of the CTMC of the whole system is never stored. Instead of that, the generator matrices of smaller subsystems are stored. With respect to time complexity, we do not solve the CTMC isomorphous to the original system but those isomorphous to the derived subsystems. The price we must pay for the above savings is that only approximate values are computed for the throughput of transitions. Experimental results on several examples generally have an error of less than 5%.

References

[1] J. Campos, J. M. Colom, H. Jungnitz, and M. Silva. Approximate throughput computation of stochastic marked graphs. *IEEE Transactions on Software Engineering*, 20(7):526–535, July 1994.

[2] J. Campos, S. Donatelli, and M. Silva. Structured solution of asynchronously communicating stochastic modules: From DSSP to general p/t systems. Research report, Departamento de Ingeniería Eléctrica e Informática, Universidad de Zaragoza, Spain, October 1997. Submitted paper.

[3] J. Campos and M. Silva. Structural techniques and performance bounds of stochastic Petri net models. In G. Rozenberg, editor, *Advances in Petri Nets 1992*, volume 609 of *Lecture Notes in Computer Science*, pages 352–391. Springer-Verlag, Berlin, 1992.

[4] J. M. Colom and M. Silva. Improving the linearly based characterization of P/T nets. In G. Rozenberg, editor, *Advances in Petri Nets 1990*, volume 483 of *Lecture Notes in Computer Science*, pages 113–145. Springer-Verlag, Berlin, 1991.

[5] F. DiCesare, G. Harhalakis, J. M. Proth, M. Silva, and F.B. Vernadat. *Practice of Petri Nets in Manufacturing*. Chapman & Hall, London, 1993.

[6] A. Gibbons. *Algorithmic Graph Theory*. Cambridge University Press, London, 1985.

[7] M. K. Molloy. Performance analysis using stochastic Petri nets. *IEEE Transactions on Computers*, 31(9):913–917, September 1982.

[8] C. J. Pérez-Jiménez, J. Campos, and M. Silva. Approximate throughput computation of a class of cooperating sequential processes. In *Proceedings of the Rensselaer's Fifth International Conference on Computer Integrated Manufacturing and Automation Technology (CIMAT'96)*, pages 382–389, Grenoble, France, May 1996.

[9] C. J. Pérez-Jiménez, J. Campos, and M. Silva. On approximate performance evaluation of manufacturing systems modelled with weighted T-systems. In *Proceedings of the IMACS/IEEE-SMC Multiconference on Computational Engineering in Systems Applications (CESA'96)*, pages 201–207, Lille, France, July 1996.

[10] C. J. Pérez-Jiménez, J. Campos, and M. Silva. State machine reduction for the approximate performance evaluation of manufacturing systems modelled with cooperating sequential processes. In *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, pages 1159–1165, Minneapolis, Minnesota, USA, April 1996.