Product form for Stochastic Automata Networks

Jean-Michel Fourneau, Brigitte Plateau, William J. Stewart

Product form for Stochastic Automata Networks

J.M. Fourneau
PRiSM CNRS UMR 8144
Universite de Versailles
Saint-Quentin
45 Av. des Etats Unis, 78000
Versailles, France
jmf@prism.uvsq.fr

B. Plateau
INRIA project MESCAL
Laboratoire Informatique de
Grenoble
CNRS UMR 5217
Montobonnot, France
Brigitte.Plateau@imag.fr

W. Stewart
North Carolina State
University,
Raleigh
North Carolina, USA
billy@csc.ncsu.edu

ABSTRACT

We consider Stochastic Automata Networks (SAN) in continuous time and we prove a sufficient condition for the steady-state distribution to have product form. We consider SAN without synchronizations where the transitions of one automaton may depend of the states of the other automata. Even with this restriction, this sufficient condition is quite simple and this theorem generalizes former results on SAN but also on modulated Markovian queues, such as the Boucherie's theory on competing Markov chain, or on reversible queues considered by Kelly. The sufficient condition and the proof are purely algebraic.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Modeling techniques; Performance attributes; G.3 [**Probability and Statistics**]: Markov processes

Keywords

Continuous Time Markov Chain, Product form steady-state distribution, Stochastic Automata Networks

1. INTRODUCTION

Since they have been introduced by B. Plateau [15] to evaluate the performance of distributed algorithms, Stochastic Automata Networks (SAN for short) have been associated to new research on numerical resolution of finite Markov chains. The key idea is to take into account the tensor decomposition of the transition matrix of a SAN to improve the storage of the model and the complexity of the vector-matrix product [5]. The first algorithm proposed was a numerical resolution of the steady-state distribution of the Markov chain associated to a SAN [17] using the power method. Since then, several numerical methods have been investigated.

As a SAN is a modular decomposition into automata which are connected by synchronized transitions and functions, SAN are closely related to Stochastic Process Algebra, (see for instance PEPA [10]). Therefore, new results on SAN may be easily translated into other models based on composition. The tensor decomposition of the generator has been generalized for Stochastic Petri Nets (see for instance [4]), Stochastic Process Algebra [11] and other modular specification methods as well.

After a decade of numerical analysis, some analytical results for SAN have been proved. First, B. Plateau et al. [18] have investigated SAN without synchronization. They proved that a product form steady-state distribution exists as soon as some local balance conditions are satisfied. Even without synchronization, the transitions of the automata are still dependent because of functional rates. Plateau's result is closely related to Boucherie's result on Markov chains in competition [2] and Robertazzi's theorems on Petri nets [13]. Similarly, using the same type of argument (i.e. group local balance), Sereno has proved a sufficient condition to obtain a product form solution for a PEPA model [20] without functional rates.

In [3], the authors have considered SAN without function but with a special case of synchronization denoted as limited synchronization. The usual assumption on synchronization used to define the SAN methodology was the ”Rendez-Vous” between an arbitrary number of automata with arbitrary transitions. In a limited synchronization, only two automata are really active. The effects of such a synchronization on other automata in the network are simple loops. The authors had proved a sufficient condition to have a product form steady-state distribution: existence of a solution for a fixed-point system between the instantaneous arrival rate and the steady-state distributions of the automata in isolation. Some typical queueing networks such as Jackson's networks or Gelenbe's networks of positive and negative customers [9] are examples of this type of SAN. For both networks, the fixed-point system is equivalent to the well-known flow equation.

In [7], one of us had considered a completely different kind of synchronization: the Domino synchronization. Briefly, a Domino synchronization is an ordered list of tuples (automaton number, list of transitions inside this automaton). The synchronization takes place according to the order in the list. The synchronization may completely succeed or be only partial if some conditions are not satisfied. Again it is proved that the existence of a solution to some fixed point system is a sufficient condition to obtain a product form solution.

Here we consider again SAN without synchronizations and we generalize the results obtained in [18]. We consider SANs without synchronization. This does not imply that the au-

Valuetools '07, October 23-25, 2007, Nantes, France
Copyright 2007 ICST 978-963-9799-00-4.

tomata are independent because the transition rate in one automaton may depend on the states of others automata. This model is general enough to represent all multidimensional Markov chains where the rates are dependent of the states of one or several components. This is typically the case of Competing Markov chains considered by Boucherie [2], some Petri nets modeled by Robertazzi [13], and modulated Markov chains. We obtain a very simple algebraic condition using an elementary algebraic proof. This assumption does not require that the local behaviors will always be irreducible for any value of the modulating states. The only requirement is the global irreducibility of the Markov chain and the existence of commons eigenvectors in the kernels of sets of matrices. We also show that this property is somehow related to some important concepts already known in queueing networks theory like reversibility and local balance property.

The rest of the paper is organized as follows: in section II we present a brief introduction to SAN and we show the universality of this description to represent multidimensional modulated Markov chains. Section III is devoted to the proof of the sufficient condition of product form while in section IV we show the links to known results and we present new product theorems based on our necessary condition.

2. INTRODUCING SAN

A *Stochastic Automata Network* (SAN) consists of a number of individual stochastic automata that operate more or less independently of each other. Each individual automaton, $\mathcal{A}$, is represented by a number of states and rules that govern the manner in which it moves from one state to the next. The state of an automaton at any time t is just the state it occupies at time t and the state of the SAN at time t is given by the state of each of its constituent automata.

We first consider a set of non-interacting automata. Then we add the synchronizations and functions to represent the dependence between automata. As our approach here is limited to SAN without synchronizations we put more emphasis on functions.

2.1 Non-Interacting Stochastic Automata

We present an example rather than the general theory which can be found now in many publications [6, 5, 1, 16, 17].

EXAMPLE 1. *Consider the case of a system that may be modelled by two completely independent stochastic automata, each of which may be represented by a continuous-time Markov chain. Let us assume that the first automaton, denoted $\mathcal{A}^{(1)}$, has n_1 states and transition rate matrix given by $Q^{(1)} \in \mathcal{R}^{n_1 \times n_1}$. Similarly, let $\mathcal{A}^{(2)}$ denote the second automaton; n_2, the number of states in its representation and $Q^{(2)} \in \mathcal{R}^{n_2 \times n_2}$, its transition rate matrix. The state of the overall (two-dimensional) system may be represented by the pair (i, j) where $i \in \{1,\ 2,\ \ldots,\ n_1\}$ and $j \in \{1,\ 2,\ \ldots,\ n_2\}$, and the infinitesimal generator of the two-dimensional system is given by $Q^{(1)} \oplus Q^{(2)}$ where $\oplus$ is the tensor sum. Remember that with*

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix},$$

the tensor sum $C = A \oplus B$ is given by

$$\left(\begin{array}{ccc|ccc} a_{11}+b_{11} & b_{12} & b_{13} & a_{12} & 0 & 0 \\ b_{21} & a_{11}+b_{22} & b_{23} & 0 & a_{12} & 0 \\ b_{31} & b_{32} & a_{11}+b_{33} & 0 & 0 & a_{12} \\ \hline a_{21} & 0 & 0 & a_{22}+b_{11} & b_{12} & b_{13} \\ 0 & a_{21} & 0 & b_{21} & a_{22}+b_{22} & b_{23} \\ 0 & 0 & a_{21} & b_{31} & b_{32} & a_{22}+b_{33} \end{array}\right).$$

Now given N *independent* stochastic automata, $\mathcal{A}^{(1)}$, $\mathcal{A}^{(2)}$, …, $\mathcal{A}^{(N)}$, with associated infinitesimal generators, $Q^{(1)}$, $Q^{(2)}$, … , $Q^{(N)}$, and probability distributions $\pi^{(1)}(t)$, $\pi^{(2)}(t)$, …, $\pi^{(N)}(t)$ at time t, the infinitesimal generator of the $N-$dimensional system, which we shall refer to as the *global generator*, is given by

$$Q = \oplus_{k=1}^{N} Q^{(k)} = \sum_{k=1}^{N} I_{n_1} \otimes \cdots \otimes I_{n_{k-1}} \otimes Q^{(k)} \otimes I_{n_{k+1}} \otimes \cdots \otimes I_{n_N} \tag{1}$$

where $\otimes$ is the tensor product. The probability that the system is in state $(i_1,\ i_2,\ \ldots,\ i_N)$ at time t, where i_k is the state of the k^{th} automaton at time t with $1 \leq i_k \leq n_k$ and n_k is the number of states in the k^{th} automaton, is given by $\prod_{k=1}^{N} \pi_{i_k}^{(k)}(t)$ where $\pi_{i_k}^{(k)}(t)$ is the probability that the k^{th} automaton is in state i_k at time t. Furthermore, the probability distribution of the N-dimensional system, $\pi(t)$, is given by the tensor product of the probability vectors of the individual automaton at time t, i.e.,

$$\pi(t) = \otimes_{k=1}^{N} \pi^{(k)}(t). \tag{2}$$

To solve N-dimensional systems that are formed from independent stochastic automata is therefore very simple. It suffices to solve for the probability distributions of the individual stochastic automata and to form the tensor product of these distributions. This resolves the case of independent stochastic automata, and we now turn our attention to automata that interact with each other

2.2 Interacting Stochastic Automata

There are two ways in which stochastic automata interact:

1. The rate at which a transition occurs may be a *function* of the state of a set of automata. Such transitions are called *functional* transitions. Transitions that are not functional are said to be *constant.*

2. A transition in one automaton may *force* a transition to occur in one or more other automata. We allow for both the possibility of a *master/slave* relationship, in which an action in one automaton (the master) actually occasions a transition in one or more other automata (the slaves), and for the case of a *rendez-vous* in which the presence (or absence) of two or more automata in designated states causes (or prevents) transitions to occur. We refer to such transitions collectively under the name of *synchronized* transitions. Synchronized transitions are triggered by a synchronizing event; indeed, a single synchronizing event will generally cause multiple synchronized transitions. Transitions that are not synchronized are said to be *local.*

The elements in the matrix representation of any single stochastic automaton are either constants, i.e., nonnegative real numbers, or functions from the global state space to the nonnegative reals. Transition rates that depend only on the state of the automaton itself, and not on the state of any other automaton, are to all intents and purposes, constant transition rates. A synchronized transition may be either functional or constant. The same is true for local transitions.

Consider as an example, a simple queueing network consisting of two service centers in tandem and an arrival process that is Poisson at rate λ. Each service center consists of an infinite queue and a single server. The service time distribution of the first server is assumed to be exponential at fixed rate μ, while the service time distribution at the second is taken to be exponential with a rate ν that varies with the number and distribution of customers in the network. Since a state of the network is completely described by the pair (n_1, n_2) where n_1 denotes the number of customers at station 1 and n_2 the number at station 2, the service rate at station 2 is more properly written as $\nu(n_1, n_2)$.

We may define two stochastic automata $\mathcal{A}^{(1)}$ and $\mathcal{A}^{(2)}$ corresponding to the two different service centers. The state space of each is given by the set of nonnegative integers $\{0,\ 1,\ 2,\ \ldots,\ \}$ since any nonnegative number of customers may be in either station. Transitions in $\mathcal{A}^{(2)}$ depend on the first automaton in two ways. Firstly the rate at which customers are served in the second station depends on the number of customers in the network and hence, in particular, on the number at the first station. Thus $\mathcal{A}^{(2)}$ contains functional transition rates, $(\nu(n_1,\ n_2))$. Secondly, when a departure occurs from the first station, a customer enters the second and therefore instantaneously forces a transition to occur within the second automaton. The state of the second automaton is instantaneously changed from n_2 to n_2+1. This entails transitions of the second type, namely synchronized transitions. The event, "departure from station 1", is a synchronizing event.

2.3 Building Generators using Functional Transitions

We return to the two original automata given in example 1 and consider what happens when one of the transition rates of the second automaton becomes a functional transition rate. Suppose, for example, that the rate of transition from state 2 to state 3 in the second automaton is $\hat{\mu}_2$ when the first automaton is in state 1 and $\tilde{\mu}_2$ when the first automaton is in state 2. The global infinitesimal generator is now ($*$ is the normalization):

$$\left(\begin{array}{ccc|ccc} * & \mu_1 & 0 & \lambda_1 & 0 & 0 \\ 0 & * & \hat{\mu}_2 & 0 & \lambda_1 & 0 \\ \mu_3 & 0 & * & 0 & 0 & \lambda_1 \\ \hline \lambda_2 & 0 & 0 & * & \mu_1 & 0 \\ 0 & \lambda_2 & 0 & 0 & * & \tilde{\mu}_2 \\ 0 & 0 & \lambda_2 & \mu_3 & 0 & * \end{array}\right).$$

If, in addition, the rate at which the first automaton produces transitions from state 1 to state 2 is $\bar{\lambda}_1$, $\hat{\lambda}_1$ and $\tilde{\lambda}_1$ depending on whether the second automaton is in state 1, 2 or 3, the two-dimensional infinitesimal generator is given by

$$\left(\begin{array}{ccc|ccc} * & \mu_1 & 0 & \bar{\lambda}_1 & 0 & 0 \\ 0 & * & \hat{\mu}_2 & 0 & \hat{\lambda}_1 & 0 \\ \mu_3 & 0 & * & 0 & 0 & \tilde{\lambda}_1 \\ \hline \lambda_2 & 0 & 0 & * & \mu_1 & 0 \\ 0 & \lambda_2 & 0 & 0 & * & \tilde{\mu}_2 \\ 0 & 0 & \lambda_2 & \mu_3 & 0 & * \end{array}\right).$$

It is known for a long time that the introduction of functional transition rates has no effect on the *structure* of the global transition rate matrix other than when functions evaluate to zero in which case a degenerate form of the original structure is obtained. However, even if the structure is preserved, the actual values of the nonzero elements prevents us from writing the solution in the simple form of equation (2). Nevertheless it is still possible to profit from this unaltered nonzero structure. This is the concept behind the extended (generalized) tensor algebraic approach, [16]. The descriptor is still written as in equation (1), but now the elements of $Q_j^{(i)}$ may be functions. This means that it is necessary to track elements that are functions and to substitute (or recompute) the appropriate numerical value each time the functional rate is needed.

2.4 notation

The following notations will be used in the paper.

- The number of automata is N.
- i, and j will be state indices
- l will be an automaton index
- m will be a matrix index.
- $Q^{(l)}$ is the transition rate matrix which describes the local transition of automaton l.
- $\vec{k}$ is a state of the Markov chain. k_l is the state of automata l when the global state is $\vec{k}$. $\vec{k}$ is a N component vector.
- $\vec{k}|l$ is the $N-1$ component vector obtained from $\vec{k}$ after deletion of component l.
- $\vec{k}+(l,i)$ is the N component vector obtained from $\vec{k}$ after replacing the value of component l by i.

We assume that the SAN contains functions. So all the matrices are functional and we note $Q^l[i,j](\vec{k},\vec{k}')$ the entry of matrix Q^l for a transition from i to j in automaton l when the global state is $\vec{k}$ and it changes to $\vec{k}'$ due to the transition.

3. PRODUCT FORM OF SAN WITHOUT SYNCHRONIZATION

Assume that the steady-state distribution has a product form ($Pr(x_0,..,x_n) = C\pi_1(x_1)...\pi_n(x_n)$). We investigate some sufficient condition for such a condition to exist but we do not provide algorithm to compute the normalization constraint C.

In this paper we introduce a simple property which, to the best of our knowledge, has not been considered before, at least in the context of steady-state distribution of CTMC: a positive vector which belongs to the kernels of all the transition rate matrices. Note that in the following we use the word kernel for left kernel or left nullspace.

DEFINITION 1. *Let α be a probability distribution. We note by $\mathcal{S}(\alpha)$ the set of transition rate matrices M such that $\alpha M = 0$ (i.e. α is in the the kernel of all matrices in $\mathcal{S}(\alpha)$).*

PROPERTY 1. *$\mathcal{S}(\alpha)$ has a several interesting properties:*

1. $\mathbf{0}$ *(the matrix whose elements are all zero) is in* $\mathcal{S}(\alpha)$
2. $aM1$ *is in* $\mathcal{S}(\alpha)$*. for all matrices* $M1$ *in* $\mathcal{S}(\alpha)$ *and* a *in* R^+*.*
3. $aM1 + bM2$ *is in* $\mathcal{S}(\alpha)$ *for all matrices* $M1$ *and* $M2$ *in* $\mathcal{S}(\alpha)$ *and* a, b *in* R^+ *such that* $a + b = 1$*.*

We will see that Boucherie's theory of competing chains is somehow linked to property 1, that the key idea of our previous work on SAN is property 2 while the algebraic interpretations of local balance and reversibility are connected to property 3.

To introduce the assumptions and the proofs, we first consider the Chapman-Kolmogorov equation for steady-state:

$$Pr(\vec{k}) \quad \left[\sum_{l=1}^{n}\sum_{i\neq k_l} Q^{(l)}[k_l,i](\vec{k},\vec{k}+(l,i))\right] = \sum_{l=1}^{n}\sum_{i\neq k_l} Q^{(l)}[i,k_l](\vec{k}+(l,i),\vec{k})Pr(\vec{k}+(l,i)). \qquad (3)$$

For the right rand side, we simply consider all the transitions into $\vec{k}$ resulting from any movement of automaton l which moves to state k_l. For the left hand side we write all the transitions leaving state $\vec{k}$ due to a movement of automaton l which jumps out of state k_l. Note that we do not have dummy transitions because the summation on state i which differs from k_l.

To improve the previous results already published on this subject it is necessary to change the way we represent functions. Remember that we consider discrete state spaces, thus the functions argument are denumerable. For any function we have in the model, we only have a denumerable set of values reached by this function. Instead of using a function terminology, we evaluate the functions and the matrices and put the matrices in a set and use an index. Let us now introduce this idea more formally.

Definition 2. *Let* l *be an automaton index, we consider all the functions in matrix* $Q^{(l)}$ *and we evaluate them for all state* $\vec{k}$ *when the transition from* $\vec{k}$ *to* $\vec{k} + (l, i)$ *takes place. Such a matrix will be denoted by* $L^{(l,m(\vec{k}))}$ *where* $m(\vec{k})$ *is an index. The set of matrices* $L^{(l,m(\vec{k}))}$ *will be denoted by* $\mathcal{F}_{(l)}$*.*

Let us first present a small model which may have product form solution for some particular functions.

Example 2. *We consider a network with two automata* $\mathcal{A}_1$ *and* $\mathcal{A}_2$*. Both have a very simple state space:* $\{0, 1\}$ *(see figure 1). The transitions in* $\mathcal{A}_1$ *have a fixed rate* $l1$ *for the transition from* 0 *to* 1 *and* $l2$ *for the transition from* 1 *to* 0*. Automaton* $\mathcal{A}_2$ *has two functional transitions: the rate from* 0 *to* 1 *has a functional rate* $f1$ *and the reverse transitions has functional rate* $f2$*. Both functions use the state of automaton* $\mathcal{A}_1$ *as an argument.*

We assume that the functions are $f1(x1) = mb + m(1 - b)1_{x1=0}$ *and* $f2(x1) = m1 + m21_{x1=0}$ *where* $x1$ *is the current state of automaton* $\mathcal{A}_1$*,* 1_x *is an indicator function and* b *is a positive values smaller than* 1*. Thus:*

- *if automaton* $\mathcal{A}_1$ *is in state* 0*, the rates are respectively* m *and* $m1 + m2$ *for the transition from* 0 *to* 1 *and for the reverse transition.*

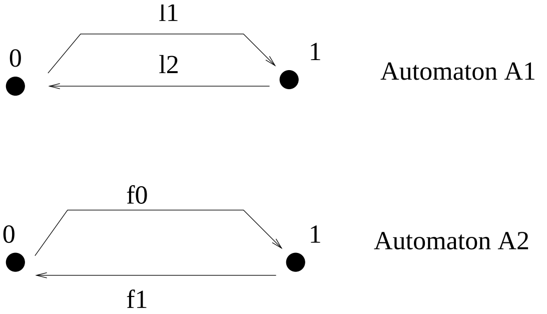

Figure 1: A SAN with functions $f0$ and $f1$ and constant rates $l0$ and $l1$.

- *if automaton* $\mathcal{A}_1$ *is in state* 0*, the rates are* mb *and* $m1$*.*

We depict in figure 2 the Markov chain of this SAN and we now compute the two matrices associated to automaton $\mathcal{A}_2$*.*

$$M0 = \begin{pmatrix} -m & m \\ m1+m2 & -m1-m2 \end{pmatrix} \textit{ and } M1 = \begin{pmatrix} -m\ b & m\ b \\ m1 & -m1 \end{pmatrix}$$

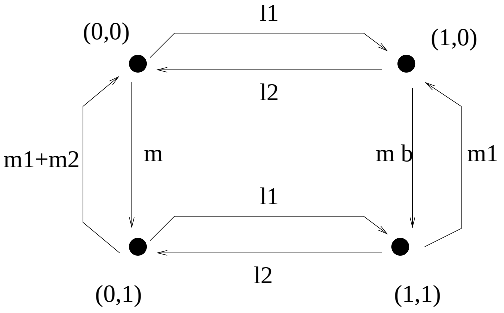

Figure 2: The Markov chain of the SAN.

The kernel of $M0$ *(resp.* $M1$*) contains all the vectors multiple of* $(m1 + m2, m)$ *(resp.* $(m1, m\ b)$*).*

We now state and prove the main results about product form for a SAN without synchronization. We assume that the Markov chain is irreducible and positive recurrent. Thus a steady state distribution exists. Note that as the assumptions we made on the matrices are very weak, they do not prove irreducibility of the Markov chain. For some particular cases however the assumptions on the local matrices implies the irreducibility of the chain.

Theorem 1. *Consider a SAN with functions but without synchronizations. Assume that the steady state exists. If for each automaton* l *there exists a probability distribution* π_l *such that all the matrices in* $\mathcal{F}_{(l)}$ *are in* $\mathcal{S}(\pi_l)$*, then the SAN has a product form steady state distribution such that:*

$$Pr(x_0, .., x_n) = C\pi_1(x_1)\dots\pi_l x_l \pi_n(x_n).$$

Proof: First we write a new version of the global balance equation after substitution of the new index for functions:

$$
\begin{aligned}
Pr(\vec{k}) \quad & \left[\sum_{l=1}^{n} \sum_{i \neq k_l} L^{(l,m(\vec{k}))}[k_l, i] \right] = \\
& \sum_{l=1}^{n} \sum_{i \neq k_l} L^{(l,m(\vec{k}))}[i, k_l] Pr(\vec{k} + (l,i)) \quad .
\end{aligned}
\tag{4}
$$

We divide both sides of the equation by $Pr(\vec{k})$ and we take into account the usual simplification rule:

$$\frac{Pr(\vec{k} + (l,i))}{Pr(\vec{k})} = \frac{\pi_l(i)}{\pi_l(k_l)}.$$

Thus we get:

$$
\begin{aligned}
& \sum_{l=1}^{n} \quad \left[\sum_{i \neq k_l} L^{(l,m(\vec{k}))}[k_l, i] \right] \\
= \quad & \sum_{l=1}^{n} \left[\sum_{i \neq k_l} L^{(l,m(\vec{k}))}[i, k_l] \frac{\pi_l(i)}{\pi_l(k_l)} \right] \quad .
\end{aligned}
$$

But $L^{(l,m(\vec{k}))}$ is a transition rate matrix and for every k_l we have:

$$\sum_{i \neq k_l} L^{(l,m(\vec{k}))}[k_l, i] = -L^{(l,m(\vec{k}))}[k_l, k_l].$$

After substitution we get:

$$
\begin{aligned}
& \sum_{l=1}^{n} \quad \left[-L^{(l,m(\vec{k}))}[k_l, k_l] \right] \\
= \quad & \sum_{l=1}^{n} \left[\sum_{i \neq k_l} L^{(l,m(\vec{k}))}[i, k_l] \frac{\pi_l(i)}{\pi_l(k_l)} \right] \quad .
\end{aligned}
\tag{5}
$$

Now remember that for all index $m(\vec{k})$, matrix $L^{(l,m(\vec{k}))}$ satisfies:

$$\pi_i L^{(l,m(\vec{k}))} = 0.$$

Therefore equation (5) is satisfied and we have proved that the SAN has a steady state distribution probability.

Corollary 1. *Consider again the SAN of example 2 matrices $M0$ and $M1$ have the same kernel if $b = \frac{m1}{m1+m2}$. If this condition is satisfied, the steady-state distribution of the SAN has product form:*

$$\pi(x1, x2) = C \left(\frac{l1}{l2} \right)^{x1} \left(\frac{m}{m1 + m2} \right)^{x2}.$$

We now present some examples and links to other theorems already published to show that we generalize many concepts and results on Markovian Petri nets, competing Markov chains, modulated set of queues and Stochastic Automata Networks. But before proceeding with this examples, let us make some remarks concerning our main theorem.

- First, it is not necessary for all the matrices to be irreducible. But the matrix resulting of SAN description must be irreducible.

Example 3. *Consider again the SAN in Fig. 1. We now change the functions $f0$ to be $m1_{x1=0}$ and $f1$ to be $m11_{x1=1}$. The global Markov chain is irreducible (see Fig. 3) but the matrices associated to automaton $\mathcal{A}_2$ are not irreducible. Indeed:*

$$M0 = \begin{pmatrix} -m & m \\ 0 & 0 \end{pmatrix} \text{ and } M1 = \begin{pmatrix} 0 & 0 \\ m1 & -m1 \end{pmatrix}.$$

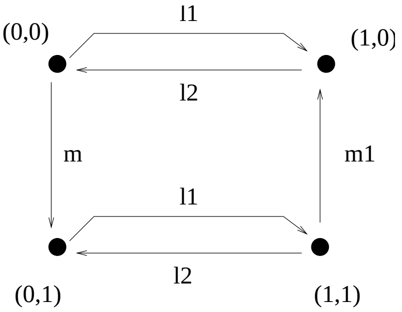

Figure 3: An irreducible Markov chain associated to a SAN whose matrices are reducible

However note that we do not claim that this SAN has a product form.

- It is possible that some matrices in the model have some rows equal to the null vector. To provide an example of such a system with product form we must consider a model based on larger state space.

Example 4. *We consider a network with two automata $\mathcal{A}_1$ and $\mathcal{A}_2$. $\mathcal{A}_2$ has a very simple state space: $\{0,1\}$ while $\mathcal{A}_1$ has four states (see Fig 4). The transitions in $\mathcal{A}_2$ have a fixed rate $l2$ for the transition from 0 to 1 and $m2$ for the transition from 1 to 0. Automaton $\mathcal{A}_1$ contains four functional transitions governed by four functions $f1$, $f2$, $f3$ and $f4$. More precisely:*

$$Q^1 = \begin{pmatrix} -l1 & l1 & 0 & 0 \\ m1 & -m1 - f1 & f1 & 0 \\ 0 & f2 & -f2 - f3 & f3 \\ 0 & 0 & f4 & -f4 \end{pmatrix}.$$

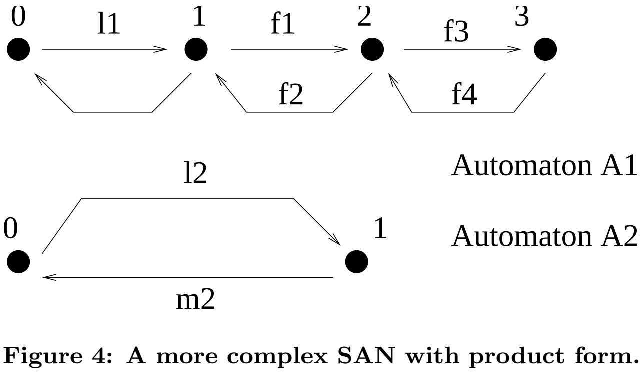

Figure 4: A more complex SAN with product form.

Let us now describe the functions:

$$\begin{cases} f1 & = & l1 & 1_{x2=1} \\ f2 & = & m1 & 1_{x2=1} \\ f3 & = & l1 & 1_{x2=0} \\ f4 & = & m1 & 1_{x2=0} \end{cases}$$

Thus the matrices associated to automaton $\mathcal{A}_1$ are:

$$M0 = \begin{pmatrix} -l1 & l1 & 0 & 0 \\ m1 & -m1 & 0 & 0 \\ 0 & 0 & -l1 & l1 \\ 0 & 0 & m1 & m1 \end{pmatrix}$$

and

$$M1 = \begin{pmatrix} -l1 & l1 & 0 & 0 \\ m1 & -m1-l1 & l1 & 0 \\ 0 & m1 & -m1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Let us now find the kernel of both matrices. For $M0$ we obtain the set of vectors

$$\{u\ (m1, l1, 0, 0) + v\ (0, 0, m1, l1), \forall\ u, v \in \mathcal{R}\}$$

while the kernel of $M1$ is

$$\{u\ (1, l1/m1, l1^2/m1^2, 0) + v\ (0, 0, 0, 1), \forall\ u, v \in \mathcal{R}\}.$$

Clearly the vector $(1, l1/m1, l1^2/m1^2, l1^3/m1^3)$ is in both sets and Theorem 1 proves that we have a product form solution.

4. LINKS TO RELATED WORKS

We first show that previous results on product form SAN satisfy the condition of theorem 1.

4.1 Previous Results on SAN

In [18] some of us have considered Stochastic Automata Networks without synchronizations and they have proved three sufficient conditions to obtain product form. As the second and third one are related to reversibility which will be presented in the next pages we only present here the first case.

THEOREM 2 (CASE 1 IN [18]). *Assume that the Stochastic Automata Network contains functional rates but no synchronization. The local transitions are described by functional matrices $L^{(l,m(\vec{k}))}$. Assume that for every automaton index l $L^{(l,m(\vec{k}))}$ is the product of a constant matrix $L^{(l)}$ by a function $f^l(\vec{k}|l)$ which does not use automaton l as an argument. Assume that matrix $L^{(l)}$ is an ergodic stochastic matrix the steady-state distribution of which is π_l. Then the steady-state distribution of the chain has product form.*

$$\pi(\vec{k}) = C \prod_l \pi_l$$

where C is a normalization constant.

The proof in [18] of this theorem is based on algebraic manipulations of the global balance equation. Note that in this case, the local matrices must be irreducible. We know show that this theorem is a simple corollary of theorem 1.

PROPERTY 2. *Assume that matrices $L^{(l,m(\vec{k}))}$ are the product of a constant matrix $L^{(l)}$ by a function $f^l(\vec{k}|l)$ which does not use the state of automaton l as an argument, then all matrices $L^{(l,m(\vec{k}))}$ associated to automaton l have the same kernel.*

Proof: For all l we have $L^{(l,m(\vec{k}))} = f^l(\vec{k}|l)L^{(l)}$. And $\pi_l L^{(l)} = 0$. Thus $\pi_l f^l(\vec{k}|l)L^{(l)} = 0$ for all $\vec{k}|l$. And after substitution $\pi_l L^{(l,m(\vec{k}))} = 0$. Therefore matrices $L^{(l,m(\vec{k}))}$ are all in set $\mathcal{S}(\pi_l)$.

Due to this property, we clearly state that Theorem 1 generalizes this first result in our previous paper. But Theorem 2 (i.e. the first case in [18] was already a generalization of a long list of results on the product form of chains which model agents competing overs resources. The typical model of such as system is the dining philosophers problem which have been shown to have product form for a long time [23, 21, 13, 22] using many modeling techniques and theoretical frameworks. The most relevant results were presented by Boucherie in [2].

4.2 Competing Markov Chain

Boucherie had introduced in [2] a new framework to generalize Petri nets with product form. This theory was denoted as competing Markov chains. He proved two theorems on product form: the first one with a more restrictive definition of resource and blocking (Th 2.6 in Boucherie's paper), and a second one where the chains in isolation must have some local balance but where the blocking over resource is weaker (Theorem 2.8 in [2]). We now present Boucherie's first theorem. The second result is mentioned in the section on partial reversibility. We need first to introduce some concepts and notations coming from Boucherie's presentation. He considered a collection of Markov chains and he introduced a product process which has a tensor representation and some restrictions on the reachable state space due to the competition over resources. For the sake of compatibility with SAN we use a tensor representation of the process which is not the original presentation by Boucherie. Let S_k be the state space of chain k. The reachable state space of the chains in competition is a subset of the Cartesian product. The transition rate matrix of the product process is the tensor sum of the transition rate matrix of the chains where some transitions have been removed. The competition over resources is modelled as an exclusion of a part of the state space S (say A) and the cancellation of some transitions. A contains the states where the resources are used by several chains. For instance, assume that automaton 1 and automaton 2 compete over resource r, if automaton 1 has resource r, all the transitions of chain 2 are cancelled. Thus the process cannot enter states in A.

For instance, Fig. 5 presents two chains $X1$ and $X2$ both with states $\{0, 1, 2, 3\}$ competing over one resource. The resource is owned by a chain when it is in state 2 or 3. It is released when the chain jumps from state 3 to 1. Thus states in $\{2, 3\} \times \{2, 3\}$ are forbidden and the reachable state space of the product process is $\{0, 1, 2, 3\} \times \{0, 1, 2, 3\}\ \{2, 3\} \times \{2, 3\}$. When process $X1$ is in state 2 or 3 process $X2$ is stopped. If process $X1$ is in state 0 or 1, process $X2$ can move. Of course we also have the symmetrical rules when $X2$ owns the resource. In Fig. 5, we have depicted first the chains $X1$ and $X2$ and the product process when the chains are competing over a resource.

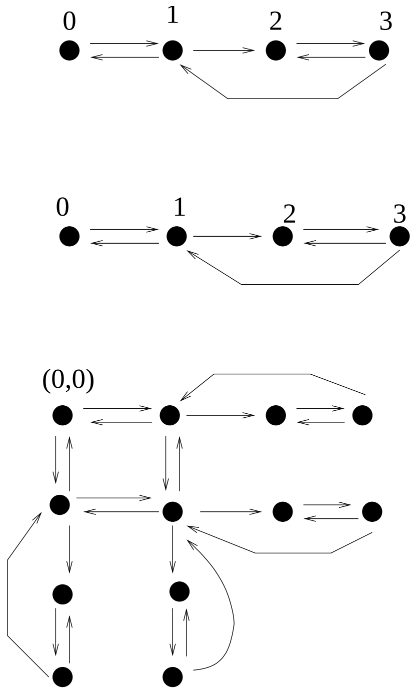

Figure 5: Two Markov chains in competition

The general definition of competing Markov chains uses a set of resources in competition I and for each resource r in I the set of chains which compete over this resource $C(r)$.

Theorem 3. *[Th 2.6 [2]] Markov chains in competition have product form (this is only a simplified presentation of Boucherie's theorem to avoid a lot of notations).*

Property 3 (SAN and Competing Markov chains). *Consider a SAN modeling a set of competing Markov chains described by Boucherie's first theorem. Let $L^{(l,m(\vec{k}))}$ be the matrix of automaton l. Then for all automaton index l, $L^{(l,m(\vec{k}))}$ is the product of a constant matrix $L^{(l)}$ by a function $f^l(\vec{k}|l)$ which does not use the state of automaton l as an argument and which takes values in $\{0,1\}$.*

Proof: following Boucherie, the transition rate of a competing Markov chain is given by the following rules:

- if states $\vec{k}$ and $\vec{k}'$ differ by more than 1 components, the transition rate is 0.
- from state $\vec{k}$ to state $\vec{k}+(l,i)$ the transition is the transition rate from k_l to i in chain l multiplied by an indicator function.
- This function is equal to zero when there exists a resource r owned by another chain which competes with l.

The first rule states that the transition matrix is a tensor sum of some matrices. The second rule says that these matrices are the original stochastic matrices of the chains multiplied by a function of the states and taking value in $\{0,1\}$. Indeed as the function is the same for all the states of competing Markov chain l (or automaton l in a SAN formulation) we can write that transition matrix $L^{(l,m(\vec{k}))}$ is the product of a constant matrix $L^{(l)}$ by a function $f^l(\vec{k}|l)$. Finally the third rule defines the functions and we can check that function $f^l(\vec{k}|l)$ do not use automaton l. As we only use indicator functions, they clearly take values in $\{0,1\}$.

Corollary 2. *Therefore Theorem 3 is a simple corollary of Theorem 2 which is itself generalized by Theorem 1.*

So we have at least obtained an algebraic formulation of BoucheriŐ’s first theorem and a generalization for arbitrary functions but our model allows to consider other results as well.

4.3 Reversibility and Partial Reversibility

We first introduce the definition of partial reversibility and show the links with the usual definition of reversibility.

Definition 3. *An ergodic Markov chain X with stochastic matrix F is partially reversible if and only if there exists a non empty subset X of the states such that for all states i and j in X we have local balance equations between i and j. More precisely let π be the steady state of F, we must have for all i,j in X,*

$$\pi(i)F(i,j) = \pi(j)F(j,i).$$

The link between partial reversibility and reversibility is clear:

Property 4. *if X is the global set of states of the Markov chain then the matrix is reversible in the usual sense (i.e. as characterized by Kelly [12], theorem 1.3 p. 6).*

As shown by Kelly, many amendments can be made to the transition rates of a reversible Markov chain such as the reversibility still holds after the modification. We will see now that we have similar properties for partially reversible chains.

Property 5 (truncation). *We now assume that the set $\mathcal{S}(\alpha)$ contains an irreducible matrix F which is partially reversible, then all matrices obtained from F after*

1. *choosing any subset of X (let call it Y)*
2. *deleting all the transitions between states of Y*

are also in $\mathcal{S}(\alpha)$.

Note that we delete the transitions but we do not delete the states. Clearly the chain is not irreducible any more.

Proof: Let F_Y the matrix obtained after deletion of transitions between states of Y. We have $F_Y + M_Y = F$ where M_Y contains the transitions between states of Y in matrix F and the normalization to be a transition rate matrix. We will prove that $\alpha F_Y = 0$. Clearly, from the construction we have:

$$\alpha F_Y = \alpha F - \alpha M_Y = -\alpha M_Y.$$

Now remember that the chain is partially reversible associated to set X. For any state i and j in X we have $\alpha(i)F(i,j) = \alpha(j)F(j,i)$. As Y is a subset of X, we have:

$$\alpha(i)F(i,j) = \alpha(i)M_Y(i,j) = \alpha(j)F(j,i) = \alpha(j)M_Y(j,i).$$

Therefore $\alpha(i)M_Y(i,j) - \alpha(j)M_Y(j,i) = 0$. Let Yj be the set Y except state j. We sum other all i in Yj to get:

$$\sum_{i \in Yj} \alpha(i)M_Y(i,j) - \alpha(j)\sum_{i \in Yj} M_Y(j,i) = 0.$$

We remark that

$$\sum_{i \in Y j} M_Y(j,i) = -M_Y(j,j)$$

as M_Y is a transition rate matrix. Finally, for all j in Y we have:

$$\sum_{i \in Y j} \alpha(i) M_Y(i,j) + \alpha(j) M_Y(j,j) = 0.$$

As $M_Y(i,j)$ is equal to 0 if i or j is not in Y, this last relation implies that $\alpha M_Y = 0$. Thus $\alpha F_Y = 0$.

Property 6. *Property 5 is a generalization of Kelly's corollary 1.10 on truncation ([12], p. 26). Indeed Kelly assumes that the chain is reversible, and that the truncated chain is irreducible, he proved that the truncated process is still reversible and that the probability distribution is the original one up to a normalization factor. When the chain in only partially reversible, the truncation described in property 5 gives the same result on the steady state distribution.*

Property 5 also generalizes the second case considered in [18] where the truncation procedure is not completely specified.

Property 7. *Let F be an irreducible transition rate matrix associated to an ergodic Markov chain. Let α be its steady-state distribution. Assume that the chain is partially reversible with set of states X, then for every subset Y of X we have for all i:*

$$\sum_{j \notin Y} (\alpha_j F(j,i) - \alpha_i F(i,j)) = 0.$$

Property 8 (multiplication). *We assume that the set $\mathcal{S}(\alpha)$ contains an irreducible matrix F which is partially reversible, then all matrices obtained from F after*

1. *choosing any subset of X (let call it Y)*
2. *multiplying all the transitions between states of Y by an arbitrary positive constant c.*

are also in $\mathcal{S}(\alpha)$.

The chain is still irreducible after the modification of the rate as $c > 0$.

Proof: again we write $F + (c-1)M_Y = F_Y$ where F_Y is the matrix obtained after modification of the rates between states of Y and M_Y is the initial transition rate matrix of the states in Y. We prove that $\alpha F_Y = 0$. As the proof is quite similar to the proof of property 5 it is omitted here.

Theorem 4. *Consider a SAN with two automata $\mathcal{A}_1$ and $\mathcal{A}_2$. Assume that:*

1. *$\mathcal{A}_1$ does not contain any functions (all the rates are constant),*
2. *for a particular state of $\mathcal{A}_1$, the matrix of $\mathcal{A}_2$ is partially reversible, let X be the set of states of $\mathcal{A}_2$ with local balance, and let Y be an arbitrary subset of X,*
3. *let R be the matrix on the states of automaton $\mathcal{A}_2$ containing the transitions between the states of Y*
4. *$\mathcal{A}_2$ contains functions whose argument is the state of $\mathcal{A}_1$, and the functions are only carried by the transition between the states of Y.*
5. *R is the product of a function f by a constant matrix $R0$*
6. *the CTMC is ergodic (one must check the irreducibility for finite chains)*

then the SAN has a product form solution.

Proof: This is a simple consequence of properties 5 and 8. Note that this is not the most general result based on these properties. We still have product form if both automata are partially reversible and the functions are designed accordingly. Note that we may prove local balance between some nodes due to structural conditions on the automata.

Definition 4 (peninsula). *Consider an ergodic CTMC associated to transition rate matrix F, a peninsula is a set of two nodes a, b such that:*

- *Removing a and b disconnects the chain and creates two connected components A and B.*
- *$a \in A$ and $b \in B$.*
- *b is the only one successor of a.*
- *a is the only one successor of b.*

A peninsula implies that we have a local balance between a and b (see equation (6)). The only transitions between A and B are the transition from a to b and the transition from b to a. The cut theorem for Markov chain states that:

$$\pi(a)F(a,c) = \pi(b)F(b,d). \tag{6}$$

The assumption on the constant c in property 8 is not necessary. We can have the same property if c is not constant for all the states in Y.

Property 9 (multiplication again). *We assume that the set $\mathcal{S}(\alpha)$ contains an irreducible matrix F which is partially reversible. We chose any subset Y of X. We build the non diagonal elements of matrix F_Y as follows:*

$$\begin{cases} F_Y(i,j) & = & c(i,j)F(i,j) & If\ i \in Y \text{ and } j \in Y \\ F_Y(i,j) & = & F(i,j) & otherwise \end{cases}$$

where we only assume that $c(i,j) = c(j,i) \geq 0$. And we add the normalization to obtain a transition rate matrix. Then F_Y is also in $\mathcal{S}(\alpha)$.

Proof: we develop αF_Y for an arbitrary state i.

- If i is not in Y then $F_Y(i,j) = F(i,j)$. Therefore component i of αF_Y is zero.
- If i is in Y, we decompose the summation:

$$\begin{aligned} \sum_j \alpha_j F_Y(j,i) &= \sum_{\substack{j \in Y \\ j \neq i}} \alpha_j F_Y(j,i) \\ &+ \alpha_i F_Y(i,i) \\ &+ \sum_{j \notin Y} \alpha_j F_Y(j,i). \end{aligned}$$

Due to the normalization, we have:

$$F_Y(i,i) = -\sum_{\substack{j\in Y\\ j\neq i}} F_Y(i,j) - \sum_{j\notin Y} F_Y(i,j).$$

Remember that i is in Y. Then if $j \in Y$ $F_Y(i,j) = c(i,j)F(i,j)$. Otherwise $F_Y(i,j) = F(i,j)$. After substitution we get:

$$\begin{aligned}\sum_j \alpha_j F_Y(j,i) &= \sum_{\substack{j\in Y\\ j\neq i}} (\alpha_j F(j,i)c(j,i) - \alpha_i F(i,j)c(i,j)) \\ &+ \sum_{j\notin Y} (\alpha_j F(j,i) - \alpha_i F(i,j)).\end{aligned}$$

The last term is zero due to property 7. And we assume that $c(i,j) = c(j,i)$. Therefore we can factorize $c(i,j)$:

$$\sum_j \alpha_j F_Y(j,i) = \sum_{\substack{j\in Y\\ j\neq i}} c(i,j)\left(\alpha_j F(j,i) - \alpha_i F(i,j)\right).$$

As i and j are members of Y, all the terms in the summation are 0. Therefore component i of αF_Y is 0.

Thus we can use several functions in the SAN instead of only one but they must satisfy the constraints of property 9. This is detailed in the next corollary.

COROLLARY 3. *If the SAN is built with the assumptions of theorem 4 but it uses several functions. If these functions verify one of the following two assumptions*

1. *the functions carried by automaton l do not use state of automaton l as an argument*
2. *if the former assumption is not true for a function f_l carried by automaton l used for the transition from i to j in automaton l, then the function must also be used from the transition from j to i and we must have: $f_l(\vec{k} + (l,i)) = f_l(\vec{k} + (l,j))$ (i.e. the function has the same value for the transition from i to j and the transition from j to i).*

then the SAN has product form.

Note that we can combine theorem 2 and 4 and corollary 3 to obtain more general assumptions on SAN with product form.

Boucherie's second theorem is a typical result one can obtain when one combine theorem 2 and 4. The key idea behind the second theorem in [2] is to assume group balance for the chain. Then the competition over a resource only takes place in some groups and is again associated to blocking of the transitions in this group when the resource is already owned. For the sake of readability we do not develop this theorem here and just give an example.

Consider the Petri net depicted in Fig. 6. This is an example of the second Boucherie's theorem. Note that this is not a model of a real problem but the result of Robertazzi's method to transform an arbitrary Petri net into a product form one.

The Markov chain associated to this Petri net is in fact depicted in Fig. 2, the SAN model is also given at the beginning of this paper and the proof of product form is already stated.

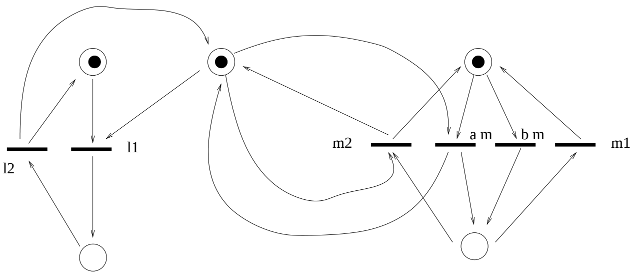

Figure 6: Example from Boucherie

4.4 Queue and set of queues in a Random environment

Single Queues in a random environment have been considered for a long time. One typically study the influence of a stochastic process (usually denoted as a phase) on the evolution of a queue or a set of queues. Following Neuts's [14] most of the results are based on the matrix geometric approach.

Only few results have been presented on open networks of n queues modulated by a phase. As the state space is usually infinite in several dimension it is not possible to apply the matrix geometric approach. In [24] Zhu had studied the steady state distribution of a modulated Markovian open network of infinite queues. He proved a sufficient condition to have a product form steady-state distribution. The steady-state distribution of the number of customers in a queue is geometrically distributed with ratio $\rho_{i,j}$ for queue i when the phase is in state j. The proof is based on the reversed process of the network of queues. The necessary condition is simple: the ratio $\rho_{i,j}$ must be constant when the phase changes. Remember that the ratios of the geometric in an open queueing network are given by the flow equations.

In [8], Verchère generalized Zhu's result as follows: he considered a multidimensional continuous-time Markov chains: the phase to model the environment and the description of the networks of queues. The transition rates of the network of queues may depend of the state of the phase but the transition rates of the phase do not change when the state of the network evolves. The chain modeling the phases is irreducible and finite. Therefore it has a steady-state distribution. Verchère first proved a general theorem on this model and he also derived corollaries when the network of queues is a G-network with positive and negative customers (see the seminal papers on positive and negative customers by Gelenbe [9]). Unlike Zhu's method, the proof in [8] is not based on reversibility.

PROPERTY 10 (VERCHÈRE'S THEOREM). *Fix the chain of the phase to an arbitrary state (say i). Let $M(i)$ be the transition rate matrix for this state of the phase. It is not necessary for $M(i)$ to be irreducible. Consider the left eigenvector problem with eigenvalue 1:*

$$\pi^i M(i) = 0,$$

if π^i does not change (up to a normalization) when the phase changes (i.e. $\pi^i = \pi$ for all phase i), the steady-state distribution of the whole system has product form $C\pi \bigotimes \alpha$ where C is a normalizing constant and α is the steady-state *distribution of the chain modelling the phase.*

Our approach clearly generalizes Verchère's result. Indeed a SAN with functions and without synchronization may represent very easily a set of queues modulated by a phase. But the SAN is even more general because we allow that the transition rates of the phase to be dependent of the state of the queues. We do not require any constraint on the functions. They may use any states of any automata to change the rate of any transition. Thus we have a much more general assumption on the model and we still have the same product-form result.

5. CONCLUDING REMARKS

We have presented a sufficient condition for a SAN without synchronization to have product form steady-state solution. This simple algebraic result generalizes many papers already published on set of queues or set of Markov chains in interaction. It remains to adapt this theory to known results on set of synchronized chains or networks of queues or general SAN. The Domino synchronization [7] have shown that we still may have product form for a complex synchronized set of automata. We also think that our approach can generalize the building blocks theory developed by Robertazzi [19] through an algebraic interpretation of product form based on local balance.

6. ACKNOWLEDGEMENTS

This work is partially supported a cooperation grant CNRS-NSF and by French ANR project (ANR-05-BLAN-009-0, Simulation and Stochastic Monotonicity).

7. REFERENCES

[1] Anne Benoit, Paulo Fernandes, Brigitte Plateau, and William J. Stewart. On the benefits of using functional transitions and kronecker algebra. *Perform. Eval.*, 58(4):367–390, 2004.

[2] Richard J. Boucherie. A characterization of independence for competing markov chains with applications to stochastic petri nets. *IEEE Trans. Software Eng.*, 20(7):536–544, also available as INRIA Report 1880., 1994.

[3] F. Boujdaine, J.M. Fourneau, and N. Mikou. Product form solution for stochastic automata networks with synchronizations, 1997. In E. Brinksma and A. Nymeyer, editors, Proc. of 5th Process Algebra and Performance Modelling Workshop.

[4] Susanna Donatelli. Superposed stochastic automata: A class of stochastic petri nets with parallel solution and distributed state space. *Perform. Eval.*, 18(1):21–36, 1993.

[5] Paolo Fernandes, Brigitte Plateau, and William J. Stewart. Optimizing tensor product computations in san. *RAIRO*, 32(3):325–351, 1998.

[6] Paulo Fernandes, Brigitte Plateau, and William J. Stewart. Efficient descriptor-vector multiplications in stochastic automata networks. *J. ACM*, 45(3):381–414, 1998.

[7] Jean-Michel Fourneau. Domino synchronization: product form solution for stochastic automata networks. *Studia Informatica*, 23(4):173–190, 2002.

[8] Jean Michel Fourneau and Dominique Vercheère. G-réseaux dans un environnement aléatoire. *RAIRO*, 34:427–448, 2000.

[9] Erol Gelenbe. Product form queueing networks with negative and positive customers. *Journal of Applied Probability*, 28:656–663, 1991.

[10] Jane Hillston. *A compositional approach to Performance Modeling.* PhD thesis, University of Edinburgh, 1994.

[11] Jane Hillston and Leïla Kloul. An efficient kronecker representation for pepa models. In Luca de Alfaro and Stephen Gilmore, editors, *PAPM-PROBMIV*, volume 2165 of *Lecture Notes in Computer Science*, pages 120–135. Springer, 2001.

[12] Franck P. Kelly. *Reversibility and Stochastic Networks.* J. Wiley, 1987.

[13] Aurel A. Lazar and Thomas G. Robertazzi. Markovian petri net protocols with product form solution. *Perform. Eval.*, 12(1):67–77, 1991.

[14] Marcel F. Neuts. *Matrix Geometric Solutions in Stochastic Models, an Algorithmic Approach.* John Hopkins University Press, 1981.

[15] Brigitte Plateau. On the stochastic structure of parallelism and synchronization models for distributed algorithms. In *SIGMETRICS*, pages 147–154, 1985.

[16] Brigitte Plateau and Jean-Michel Fourneau. A methodology for solving markov models of parallel systems. *J. Parallel Distrib. Comput.*, 12(4):370–387, 1991.

[17] Brigitte Plateau, Jean-Michel Fourneau, and K.H. Lee. Peps: A package for solving complex markov models of parallel systems. In *Proceedings of the 4th Int. Conf. on Modeling Techniques and Tools for Computer Performance Evaluation, Majorca, Spain*, pages 291–305, 1988.

[18] Brigitte Plateau and W.J. Stewart. *Advances in Computational Probability, Chapter Stochastic Automata Networks, Ed. Winfried Grassmann.* International Series in Operations Research and Management Science, Vol. 24, Kluwer Academic Publishers, 1999.

[19] Thomas G. Robertazzi. *Computer Networks and Systems: queueing networks and performance evaluation.* Springer Verlag, 1994.

[20] Matteo Sereno. Towards a product form solution for stochastic process algebras. *Comput. J.*, 38(7):622–632, 1995.

[21] Irvin Y. Wang and Thomas G. Robertazzi. Service stage petri net models with product form solution. *Queueing Syst.*, 7(3):355–373, 1990.

[22] Bernard Ycart. The philosopher's process: an ergodic reversible nearest particle system. Research report, Laboratoire de Mathématiques appliquées, Université de Pau et des pays de l'Adour, 1991.

[23] Alexandre Zenie. Colored stochastic petri nets. In *PNPM*, pages 262–271. IEEE Computer Society, 1985.

[24] Y. Zhu. Markovian queueing networks in a random environment. *Operations Research Letters*, 15:11–17, 1994.