CWI Tracts

CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
Telephone 31 - 20 592 9333, telex 12571 (mactr nl),
telefax 31 - 20 592 4199

CWI is the nationally funded Dutch institute for research in Mathematics and Computer Science.

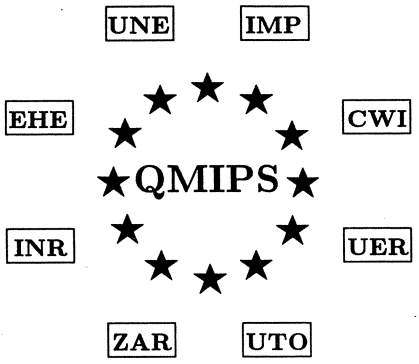
Performance evaluation of parallel and distributed systems Solution methods

Proceedings of the third QMIPS workshop
Part 2

O.J. Boxma, G.M. Koole (eds.)

1991 Mathematics Subject Classification: 60K25, 68M20.
ISBN 90 6196 445 8
NUGI-code: 811

Printed in the Netherlands



Quantitative
Modeling
In
Parallel
Systems

The QMIPS project is a collaborative research project supported by the CEC as ESPRIT-BRA project no 7269. It is being carried out by the following organisations: CWI (Amsterdam), EHEI (University of Paris V), Imperial College (London), INRIA (Sophia-Antipolis), University of Erlangen, University of Newcastle, University of Torino and University of Zaragoza.

Contents

Part 1

Part 2

Introduction

These are the proceedings of the third QMIPS workshop, held in Torino, Italy, on September 25 and 26, 1993. The QMIPS project is a collaborative research project supported by the European Union, and it is carried out by 8 organizations from 6 different European countries. It is concerned with quantitative modeling in parallel and distributed systems. Within the framework of the QMIPS project several workshops are being organized. After workshops in Sophia-Antipolis (France) on Petri nets, and in Erlangen (Germany) on modeling formalisms, this workshop focused on solution methods.

Three steps can be distinguished in the analysis of parallel or distributed systems. The first is modeling, using one of the available formalisms. Depending on the formalism used, a solution method is employed to obtain performance measures for the system. This second step is the subject of these proceedings. The third step is the optimisation of the system. Research in this area is presented at the fourth QMIPS workshop in London, on April 14 and 15, 1994.

The proceedings start with two survey papers, one on solution methods for queueing models, and one on solution methods for Petri net models. The other 16 papers, all concerned with current research topics, are divided in three parts, depending on the formalism used: queueing, Petri nets or the (Max,+) algebra.

The first formalism is queueing. The paper by ETTL AND MITRANI analyses two queueing models using the recently developed spectral expansion method. BOXMA AND VAN HOUTUM apply the compensation approach to a 2×2 switch, and MITRANI AND WRIGHT solve a two-dimensional queueing problem using the boundary value technique. The next two papers deal with queueing models with negative customers, which are customers with the ability to cancel regular customers. The paper by FOURNEAU, GELENBE AND SUROS extends product form results for regular queueing networks to networks with negative customers. The paper by HARRISON AND PITEL studies tandem models which do not have product form solutions, and analyses them using the boundary value technique. KOOLE shows that the power series algorithm, which has been applied to many queueing models, can also be used for general Markov chains.

The part on Petri nets starts with the paper by BOUCHERIE AND SERENO. It characterises product form Petri nets in terms of the structure of the net. The paper by SERENO AND BALBO considers product forms as well, but focuses on computational algorithms. Also the paper by CHIOLA, ANGLANO, CAMPOS, COLOM AND SILVA studies a technique originating from queueing, namely operational analysis, which leads them to performance bounds. FRANCESCHINIS AND MUNTZ derive performance bounds for certain Petri nets that exhibit symmetry. CAMPOS, COLOM, SILVA AND TERUEL study a model consisting of several sequential processes communicating through buffers, and derive both qualitative and quantitative results. The paper by BACCELLI AND GAUJAL is concerned with free choice Petri nets. For this class of nets qualitative properties are derived. In the paper by CAMPOS, COLOM, JUNGNITZ AND SILVA marked graphs (Petri nets where each place has only one input and output arc) are considered, and a technique is introduced to approximate the throughput of

marked graphs. CANALES AND GAUJAL also study marked graphs, and exhibit the inherent parallelism to derive efficient parallel simulation procedures.

Marked graphs are strongly related to the (Max,+) algebra, and as such the two papers on this algebra are relevant to the study of Petri nets. JEAN-MARIE studies stochastic event graphs by an analysis based on the (Max,+) algebra. MAIRESSE derives some deep results on small matrices in the (Max,+) algebra.

Finally, a few words of thanks. We thank the local organizer G. Balbo and his co-workers (University of Torino) for taking care of the local arrangements and for selecting such a wonderful location for the workshop. We should like to express our gratitude to the Centre for Mathematics and Computer Science (CWI) for its support in publishing these proceedings. We are in particular grateful to Yvonne Samseer from the CWI typesetting department for her many valuable contributions to the preparation of the final manuscript, and to managing editor Wim Aspers.

Onno Boxma and Ger Koole

A Structural Characterisation of Product Form Stochastic Petri Nets

Richard J. Boucherie*
CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
Matteo Sereno†
Università di Torino
Dipartimento di Informatica, Corso Svizzera 185, 10149 Torino, Italy

Product form results for the equilibrium distribution of stochastic Petri nets are available in the literature. These results are based on assumptions for the Markov chain describing the Petri net, and not on the structure of the Petri net. The structure of the Petri net is one of the most important parts in the analysis of Petri nets, and many results on this structure are available in the literature. Hence, it seems natural to characterise the product form property on a structural level. This paper provides such a characterisation: it gives a necessary and sufficient condition for the existence of a solution for the traffic equations (the basic equations allowing product form), completely in terms of the T-invariants of the Petri net.

1 Introduction

Performance is an important issue in the design and implementation of real life systems such as computer systems, telecommunication networks, and flexible manufacturing systems. In many theoretical and practical studies of performance models involving stochastic effects, the statistical distribution of items over places is of great interest since most of the performance measures such as throughput and utilization can be derived from this distribution. If we are interested in quantitative results we can use approximation and simulation techniques. Analytical results, however, yield vital insight into the qualitative behaviour of the system. In particular, qualitative results related to the structure of the system are of great importance.

*ERCIM fellow at INRIA from September 1st, 1992 to May 31st, 1993, and at CWI from June 1st, 1993 to February 28th, 1994. ERCIM stands for European Research Consortium for Informatics and Mathematics and comprises 10 institutes: AEDIMA (Spain), CNR (Italy), CWI (The Netherlands), FORTH (Greece), GMD (Germany), INESC (Portugal), INRIA (France), RAL (UK), SICS (Sweden), SINTEF DELAB (Norway).

†Partially supported by the Italian National Research Council "Progetto Finalizzato Sistemi Informatici e Calcolo Parallelo (Grant N. 92.01563.PF69)" and by the European Grant BRA-QMIPS of CEC DG XIII

For queueing networks an important analytical result is the *product form equilibrium distribution* for the number of customers at the stations. Product form distributions were found by Jackson [17], and are nowadays known for a fairly wide class of queueing models (e.g., Baskett *et al.* [2], Boucherie and van Dijk [4], Henderson and Taylor [15], Serfozo [23]). The obvious advantage of these product form distributions is their simplicity which makes them easy to use for computational issues as well as for theoretical reflections on performance models involving congestion as a consequence of queueing.

Recently, product form results were found for the marking process of stochastic Petri nets by Lazar and Robertazzi [18]. Although these results were shown for a very special class of stochastic Petri nets consisting only of linear task sequences, the notion of competition over resources incorporated in these models cannot be included in queueing networks without the introduction of state-dependent routing. Still, product form results very similar to those obtained by Jackson [17] were found. Since these first product form results various extensions have been found. In a number of papers, Henderson *et al.* [13], [14], [16] derive product form results for stochastic Petri nets similar to those obtained for batch routing queueing networks (Boucherie and van Dijk [4], Henderson and Taylor [15]). Frosch [11], [12] derived product form results for closed synchronized systems of stochastic sequential processes, a class of Petri nets in which state machines are synchronized via buffers.

The product form results for stochastic Petri nets are based on the assumption that a positive solution exists for a linear set of equations similar to the traffic equations for queueing networks. However, a characterisation of this assumption based on the structure of the Petri net is not available in the literature. This paper provides such a characterisation. We show that a necessary and sufficient condition for the existence of a positive solution for the traffic equations is that all transitions of the Petri net are covered by closed support T-invariants. A T-invariant is a closed support T-invariant if the firing sequence is a linear chain of transitions, that is a closed support T-invariant closely resembles the 'task sequences' used by Lazar and Robertazzi [18] to prove their product form result. As will be shown via examples, the class of Petri nets used in the present paper is substantially larger than the class of Lazar and Robertazzi.

Product form results for stochastic Petri nets of a completely different type are derived by Boucherie [3]. There the equilibrium distribution for a stochastic Petri net containing several subnets linked via buffer places is shown to be a product over the subnets under some conditions. Also, closed form expressions for the equilibrium distribution of stochastic Petri nets are derived by Florin and Natkin [9]. In that paper the equilibrium distribution of a stochastic Petri net with finite reachability set is shown to be a sum of product form distributions. The number of product form distributions in this sum is related to the number of distinct markings of the Petri net, a number that is usually substantially smaller than the cardinality of the reachability set. We do not consider these types of closed form equilibrium distributions in this paper.

In section 2 we present the basic Petri net notation. In section 3 we present

the structural characterisation of the Petri net allowing us to provide necessary and sufficient conditions for the existence of a solution for the traffic equations. We will also give some known product form theorems based on the existence of such a solution. This allows us to illustrate the results by means of some simple examples in section 4.

2 Model

This section presents the basic definitions of stochastic Petri nets. For additional results and definitions, see the recent survey of Murata [21]. The specific assumptions and definitions needed to obtain product forms for stochastic Petri nets will be given in section 3.

Definition 2.1 (Marked stochastic Petri net) *A marked stochastic Petri net is a 6-tuple*

$$\mathcal{SPN} = (P, T, I, O, R, \mathbf{m}_0),$$

where $P = \{p_1, \ldots, p_N\}$ *is a finite set of places;* $T = \{t_1, \ldots, t_M\}$ *is a finite set of transitions;* $P \cap T = \emptyset$ *and* $P \cup T \neq \emptyset$; $I, O : P \times T \to \mathbb{N}_0$ *are the input and output functions identifying the relation between the places and the transitions;* $R = (r(t_1), \ldots, r(t_M))$ *is a set of firing rates drawn from exponential distributions; and* $\mathbf{m}_0$ *is the initial marking.*

A *marking* $\mathbf{m} = (\mathbf{m}(n),\ n = 1, \ldots, N)$ of a Petri net is a vector in $\mathbb{N}_0^N$, where $\mathbf{m}(n)$ represents the number of tokens at place p_n, $n = 1, \ldots, N$.

Distributions associated with different transitions are independent, and each transition of the Petri net is due to exactly one transition $t \in T$ that fires. The *execution policy* of the stochastic Petri net is the *race model with age memory* (cf. Ajmone Marsan *et al.* [1]).

From $I(\cdot,\cdot)$ and $O(\cdot,\cdot)$ we obtain the vectors $\mathbf{I}(t) = (I_1(t), \ldots, I_N(t))$, and $\mathbf{O}(t) = (O_1(t), \ldots, O_N(t))$, where $I_i(t) = I(p_i, t)$, and $O_i(t) = O(p_i, t)$. The vectors $\mathbf{I}(t)$, and $\mathbf{O}(t)$ are called *input, and output bags* of transition $t \in T$, representing the number of tokens needed at the places to fire transition t, and the number of tokens released to the places after firing of transition t. Furthermore, define the sets of places corresponding to input and output bags of transitions as ${}^\bullet t = \{p \in P | I(p,t) > 0\}$, the set of places giving input to transition t, $t^\bullet = \{p \in P | O(p,t) > 0\}$, the set of places receiving output from transition t. If transition t is *enabled* in marking $\mathbf{m}$ and fires, then the next state of the Petri net is $\mathbf{m}' = \mathbf{m} - \mathbf{I}(t) + \mathbf{O}(t)$. Symbolically this will be denoted as $\mathbf{m}[t > \mathbf{m}'$. A necessary and sufficient condition for t to be enabled is that $\mathbf{m}(n) \geq I_n(t)$, $n = 1, \ldots, N$.

A finite sequence of transitions $\sigma = t_{\sigma_1} t_{\sigma_2} \cdots t_{\sigma_k}$ is a finite *firing sequence* of the Petri net if there exists a sequence of markings $\mathbf{m}_{\sigma_1}, \ldots, \mathbf{m}_{\sigma_k}$ for which $\mathbf{m}_{\sigma_i}[t_{\sigma_i} > \mathbf{m}_{\sigma_{i+1}}$, $i = 1, \ldots, k-1$. In this case marking $\mathbf{m}_{\sigma_k}$ is *reachable* from marking $\mathbf{m}_{\sigma_1}$ by firing σ, denoted as $\mathbf{m}_{\sigma_1}[\sigma > \mathbf{m}_{\sigma_k}$. The *reachability set* $\mathcal{M}(\mathbf{m}_0)$ is a subset of $\mathbb{N}_0^N$ and gives all possible markings of the Petri net with initial marking $\mathbf{m}_0$.

The *incidence matrix* is the $N \times M$ matrix A with entries $A(i,t) = O_i(t) - I_i(t)$ describing the change in the number of tokens in place p_i if transition t fires, $i = 1, \ldots, N$, $t \in T$. A vector $\bar{\sigma}$ is the *firing count vector* of the firing sequence σ if $\bar{\sigma}(t)$ equals the number of times transition t occurs in the firing sequence σ. If $\mathbf{m}_0[\sigma > \mathbf{m}$, then $\mathbf{m} = \mathbf{m}_0 + A\bar{\sigma}$, an equation referred to as the *state equation* for the Petri net.

A vector $\mathbf{x} \in \mathbb{N}_0^M$ is a *T-invariant* if $\mathbf{x} \neq 0$, and $A\mathbf{x} = 0$. From the state equation we obtain that a T-invariant corresponds to a firing sequence that brings a marking back to itself (Murata [21]). The *support* of a T-invariant $\mathbf{x}$ is the set of transitions corresponding to non-zero entries of $\mathbf{x}$, and is denoted by $\|\mathbf{x}\|$, i.e. $\|\mathbf{x}\| = \{t \in T | \mathbf{x}(t) > 0\}$. A T-invariant $\mathbf{x}$ is a *minimal T-invariant* if there is no other T-invariant $\mathbf{x}'$ such that $\mathbf{x}'(m) \leq \mathbf{x}(m)$ for all m. A support is minimal if no proper nonempty subset of the support is also a support of a T-invariant. From Memmi and Roucairol [19] we obtain that there is a unique minimal T-invariant corresponding to a minimal support (*minimal support T-invariant*), and any T-invariant can be written as a linear combination of minimal support T-invariants. A vector $\mathbf{y} \in \mathbb{N}_0^N$ is a *P-invariant* (sometimes called S-invariant) if $\mathbf{y} \neq 0$, and $\mathbf{y}A = 0$. P-invariants correspond to conservation of tokens in subsets of places. For example, the set of places of a Petri net corresponding to a closed Jackson network is a P-invariant. Definitions of and results for minimal support etc. are analogous to those for T-invariants.

The stochastic process describing the evolution of the Petri net is a continuous-time Markov chain with state space isomorphic to the reachability set, that is with state space $\mathcal{M}(\mathbf{m}_0)$ (Molloy [20]). The transition rates of this Markov chain are denoted by $Q = (q(\mathbf{m}, \mathbf{m}'),\ \mathbf{m}, \mathbf{m}' \in \mathcal{M}(\mathbf{m}_0))$. A collection of positive numbers, $m = (m(\mathbf{m}),\ \mathbf{m} \in \mathcal{M}(\mathbf{m}_0))$, is called an *invariant measure* if it satisfies the *global balance equations*,

$$\sum_{\mathbf{m}' \in \mathcal{M}(\mathbf{m}_0)} \{m(\mathbf{m})q(\mathbf{m}, \mathbf{m}') - m(\mathbf{m}')q(\mathbf{m}', \mathbf{m})\} = 0, \quad \mathbf{m} \in \mathcal{M}(\mathbf{m}_0).$$

When m is a proper distribution over $\mathcal{M}(\mathbf{m}_0)$ it will be called an *equilibrium distribution*, and will be denoted by $\pi = (\pi(\mathbf{m}),\ \mathbf{m} \in \mathcal{M}(\mathbf{m}_0))$.

As the Markov chain is chosen such that it describes the evolution of the stochastic Petri net under consideration, irreducibility and positive recurrence properties necessary to obtain a unique equilibrium distribution for the Markov chain should be characterised directly from the Petri net structure. A Petri net is *live* if, no matter what marking has been reached from $\mathbf{m}_0$, it is possible to ultimately fire any transition of the net by progressing through some further sequence. For unicity of the equilibrium distribution we must add the assumption that the Petri net is (strongly) connected. An extensive discussion of liveness, and related concepts is given in Murata [21].

3 Product Form Results

Without loss of generality, we may assume that the firing rate associated with transition $t \in T$ with input bag $\mathbf{I}(t)$ and output bag $\mathbf{O}(t)$ can be written as

$r(t) = \mu(t)p(\mathbf{I}(t), \mathbf{O}(t))$, a form chosen in accordance with the literature on product form results (e.g., Jackson [17], Baskett *et al.* [2]).

Assume that the stochastic Petri net can be represented by a stable and regular, continuous-time Markov chain $\mathbf{X} = \{X(t),\ t \geq 0\}$ at state space $\mathcal{M}(\mathbf{m}_0)$. Then the transition rates of $\mathbf{X}$ are

$$q(\mathbf{I}(t), \mathbf{O}(t); \mathbf{m} - \mathbf{I}(t)) = \mu(t)p(\mathbf{I}(t), \mathbf{O}(t)), \tag{1}$$

for all $t \in T$, $\mathbf{m} \in \mathcal{M}(\mathbf{m}_0)$ such that $\mathbf{m} - \mathbf{I}(t) \in \mathbb{N}_0^N$. Here $q(\mathbf{I}(t), \mathbf{O}(t); \mathbf{m} - \mathbf{I}(t))$ is the transition rate associated with transition t bringing $\mathbf{m}$ to $\mathbf{m} - \mathbf{I}(t) + \mathbf{O}(t)$. The total transition rate from $\mathbf{m}$ to $\mathbf{m}' = \mathbf{m} - \mathbf{I}(t) + \mathbf{O}(t)$ is $q(\mathbf{m}, \mathbf{m}') = \sum_{\{\mathbf{n},\ t \in T: \mathbf{n} + \mathbf{I}(t) = \mathbf{m},\ \mathbf{n} + \mathbf{O}(t) = \mathbf{m}'\}} q(\mathbf{I}(t), \mathbf{O}(t); \mathbf{n})$.

Let $\mathbf{x}^1, \ldots, \mathbf{x}^h$ denote the minimal support T-invariants found from the incidence matrix. The following definition and assumption are essential to the analysis presented in this paper. Closedness of T-invariants was first defined by Donatelli and Sereno [8] as a unifying principle to obtain product form distributions for stochastic Petri nets. A necessary condition for a product form equilibrium distribution similar to closedness is presented in Henderson *et al.* [13], Corollary 1.

DEFINITION 3.1 (CLOSED SET) *For $\mathcal{T} \subset T$ define $\mathcal{R}(\mathcal{T})$, the set of input and output bags for the transitions in $\mathcal{T}$, as*

$$\mathcal{R}(\mathcal{T}) = \bigcup_{t \in \mathcal{T}} \{\mathbf{I}(t) \cup \mathbf{O}(t)\}.$$

$\mathcal{T}$ is a closed set if for any $\mathbf{g} \in \mathcal{R}(\mathcal{T})$ there exist t, $t' \in \mathcal{T}$ such that $\mathbf{g} = \mathbf{I}(t)$, as well as $\mathbf{g} = \mathbf{O}(t')$, that is if each output bag is also an input bag for a transition in $\mathcal{T}$.

ASSUMPTION 3.2 (MINIMAL CLOSED SUPPORT T-INVARIANTS) *Assume that all transitions $t \in T$ are covered by minimal closed support T-invariants, that is assume that for all $t \in T$ there exists an $i \in \{1, \ldots, h\}$ such that $t \in \|\mathbf{x}^i\|$ and $\|\mathbf{x}^i\|$ is a closed set.*

Observe that the essential part of the assumption is that all transitions are contained in a *closed* support. The assumption that all transitions are covered by minimal support T-invariants (closed or not closed) is a natural assumption if we are interested in the equilibrium or stationary distribution of a stochastic Petri net. If this assumption is not satisfied, then there exists a transition, say t_0, that is enabled in a reachable marking $\mathbf{m}$, and $t_0 \notin \bigcup_{i=1}^h \|\mathbf{x}^i\|$ (if t_0 is never enabled, then we can delete t_0 from T). Let t_0 fire in marking $\mathbf{m}$. Then there exists no firing sequence from $\mathbf{m} - \mathbf{I}(t_0) + \mathbf{O}(t_0)$ back to $\mathbf{m}$ (otherwise t_0 would be contained in a T-invariant). Thus $\mathbf{m}$ is a transient state and does not appear in the equilibrium description of the stochastic Petri net. As a consequence, both $\mathbf{m}$ and t_0 can be deleted from the equilibrium description of the Petri net.

The structural characterisation of product form results for stochastic Petri nets is completely based on Assumption 3.2. We now proceed with a characterisation of minimal *closed* support T-invariants. This shows the relation between

minimal closed support T-invariants and 'task sequences' (corresponding to a number of tasks that must be executed consecutively) as introduced by Lazar and Robertazzi [18]. This turns out to be a key-notion when product form equilibrium distributions are desired.

THEOREM 3.3 *Assume that* $\mathbf{x}$ *is a minimal closed support* T*-invariant. Then the firing sequence of* $\mathbf{x}$ *is 'linear', that is for each* $t \in \|\mathbf{x}\|$ *there is a unique* $t' \in \|\mathbf{x}\|$ *such that* $\mathbf{O}(t) = \mathbf{I}(t')$. *As a consequence* $x_i \leq 1$, $i = 1, \ldots, M$. *Conversely, if the firing sequence of a* T*-invariant* $\mathbf{x}$ *is linear, then* $\mathbf{x}$ *is a closed support* T*-invariant.*

Proof Let $t \in \|\mathbf{x}\|$. The existence of $t' \in \|\mathbf{x}\|$ such that $\mathbf{O}(t) = \mathbf{I}(t')$ follows from the closedness of $\|\mathbf{x}\|$. To proof the unicity, let $t \in \|\mathbf{x}\|$, and $t', t'' \in \|\mathbf{x}\|$ such that $\mathbf{O}(t) = \mathbf{I}(t') = \mathbf{I}(t'')$. Without loss of generality, assume that $\mathbf{O}(t') \neq \mathbf{O}(t'')$ (otherwise $t' = t''$). As a consequence there exists a place $p = p_i$ such that $\max\{O_i(t'), O_i(t'')\} - \min\{O_i(t'), O_i(t'')\} \neq 0$. Without loss of generality, assume that $O_i(t') > O_i(t'')$.

From the closedness of $\|\mathbf{x}\|$ we obtain that there exist two *distinct* transitions, say $t_1' \in \|\mathbf{x}\|$, and $t_1'' \in \|\mathbf{x}\|$ such that $\mathbf{O}(t') = \mathbf{I}(t_1')$, and $\mathbf{O}(t'') = \mathbf{I}(t_1'')$, and we must have one of the following three situations:

(a) $\mathbf{O}(t_1') = \mathbf{O}(t_1'')$, but this implies that there exist two firing sequences within $\|\mathbf{x}\|$ that can fire independently from $\mathbf{I}(t)$ to $\mathbf{O}(t_1')$, in contrast with the assumption that $\mathbf{x}$ is a minimal T-invariant.

(b) $\exists\ p' = p_j$ such that $\max\{O_j(t_1'), O_j(t_1'')\} - \min\{O_j(t_1'), O_j(t_1'')\} \neq 0$, and $O_j(t_1') > O_j(t_1'')$. This is the situation observed when we considered t' and t'' and is either followed by situation (a), (b), or (c).

(c) as (b), but now $O_j(t_1') > O_j(t_1'')$. It is obvious that this is followed by (a), (b), or (c) as well.

Finally, since $\mathbf{x}$ is a T-invariant, it must be that the firing sequences starting with t' and t'', say $t't_1' \cdots t_{\alpha'}'$ and $t''t_1'' \cdots t_{\alpha''}''$, are such that $\mathbf{O}(t_{\alpha'}') = \mathbf{O}(t_{\alpha''}'')$ for some α', α'', that is situation (a) must occur finally, which contradicts the assumption that $\mathbf{x}$ is a minimal T-invariant, because we have created two firing sequences that can independently be fired from $\mathbf{I}(t)$ to $\mathbf{O}(t_{\alpha'}')$. This establishes unicity.

Unicity implies that each transition $t \in \mathbf{x}$ can occur at most once in the firing sequence associated with $\mathbf{x}$, i.e. that $x_i \leq 1$, $i = 1, \ldots, M$.

If the firing sequence of a T-invariant $\mathbf{x}$ is linear, then for each $t \in \|\mathbf{x}\|$ there exist $s, s' \in \|\mathbf{x}\|$ such that $\mathbf{O}(s) = \mathbf{I}(t)$, $\mathbf{O}(t) = \mathbf{I}(s')$ implying that $\mathbf{x}$ has closed support. □

The important property of closed support T-invariants with respect to product form results is that the residual marking of tokens that remain at the places during one complete firing of the T-invariant is the same for all transitions, that is the firing sequence can be represented by the sequence of markings

$\mathbf{m} = \mathbf{n}+I(t_{i_1}) \rightarrow \mathbf{n}+I(t_{i_2}) \rightarrow \cdots \rightarrow \mathbf{n}+I(t_{i_k}) \rightarrow \mathbf{n}+I(t_{i_1})$, with $\mathbf{n} \equiv \mathbf{m}-\mathbf{I}(t_{i_1})$ the residual marking. This observation is the basis of the classification of the transitions into *equivalence classes* as presented below. This classification is based on a classification presented in Frosch [10], Frosch and Natarajan [11] for cyclic state machines. In the case of cyclic state machines the input bag of a transition basically contains only one place, whereas the generalisation to closed support T-invariants incorporates more general input bags. The classification will then be used to construct a solution to the traffic equations, a set of linear equations defined by analogy with the traffic equations for queueing networks.

DEFINITION 3.4 (TRAFFIC EQUATIONS) *For $t \in T$, an invariant measure, $y = (y(\mathbf{I}(t)),\ t \in T)$, for the traffic equations is a mapping $y : \mathbb{N}_0^N \rightarrow \mathbb{R}^+$ that satisfies the traffic equations for all $t \in T$ (recall the definition of the transition rates (1))*

$$\sum_{t' \in T} \{y(\mathbf{I}(t))\mu(t)p(\mathbf{I}(t), \mathbf{I}(t')) - y(\mathbf{I}(t'))\mu(t')p(\mathbf{I}(t'), \mathbf{I}(t))\} = 0. \tag{2}$$

REMARK 3.5 (TRAFFIC EQUATIONS) The definition of the traffic equations relies heavily on the assumption that all transitions are covered by closed support T-invariants. Otherwise $p(\mathbf{I}(t), \mathbf{I}(t'))$ may be zero for all $t' \in T$ since without the assumption of closedness $\mathbf{O}(t)$ need not be an input bag for some transition t'. In fact, from Assumption 3.2 we obtain that for each t there exists a t' such that $\mathbf{O}(t) = \mathbf{I}(t')$, and the first summation in the traffic equations is equivalent to $\sum_{O(t) \in \mathbf{N}_0^N} y(\mathbf{I}(t))\mu(t)p(\mathbf{I}(t), \mathbf{O}(t))$. Obviously, the second summation is equivalent to $\sum_{I(t') \in \mathbf{N}_0^N : \mathbf{O}(t') = \mathbf{I}(t)} y(\mathbf{I}(t'))\mu(t')p(\mathbf{I}(t'), \mathbf{O}(t'))$, which shows that under Assumption 3.2 the traffic equations do not exclude any transitions depositing or consuming $\mathbf{I}(t)$. In particular, Assumption 3.2 implies that the traffic equations are equivalent to the global balance equations for the Markov chain with transition rates (1), a result used below to prove that Assumption 3.2 is necessary and sufficient for the existence of a solution for the traffic equations. □

We will now show that Assumption 3.2 is necessary and sufficient for the existence of an invariant measure for the traffic equations (2). Before proving this result we first characterise the minimal support T-invariants that are connected as (2) decomposes into disjoint sets of equations, one set of equations for each equivalence class of connected T-invariants.

Assume that the minimal support T-invariants $\mathbf{x}^1, \ldots, \mathbf{x}^h$ are numbered such that $ClT \stackrel{\text{def}}{=} \{\mathbf{x}^1, \ldots, \mathbf{x}^k\}$ is the set of minimal closed support T-invariants $(k \leq h)$.

DEFINITION 3.6 (COMMON INPUT BAG RELATION) *Let $\mathbf{x}, \mathbf{x}' \in ClT$. We say that $\mathbf{x}, \mathbf{x}'$ are in common input bag relation (notation: $\mathbf{x}\ CI\ \mathbf{x}'$) if there exist $t \in \|\mathbf{x}\|$, $t' \in \|\mathbf{x}'\|$ such that $\mathbf{I}(t) = \mathbf{I}(t')$. The relation CI^* is the transitive closure of CI.*

The transitive closure of a relation is defined as follows: if $\mathbf{x}$, $\mathbf{x}'$, $\mathbf{x}'' \in \mathcal{C}lT$, and $\mathbf{x}\ CI\ \mathbf{x}'$, $\mathbf{x}'\ CI\ \mathbf{x}''$, then we define $\mathbf{x}\ CI^*\ \mathbf{x}'$, $\mathbf{x}'\ CI^*\ \mathbf{x}''$, *and* $\mathbf{x}\ CI^*\ \mathbf{x}''$. This reflects the property that we can go from $\mathbf{x}$ to $\mathbf{x}''$ via $\mathbf{x}'$. This makes the common input bag relation CI^* an equivalence relation on $\mathcal{C}lT$.

The common input bag relation characterises the irreducible sets of the Markov chain $\mathbf{Y} = (Y(t),\ t \geq 0)$ at finite state space $S = \{\mathbf{I}(t),\ t \in T\}$ with transition rates $q(\mathbf{I}(t), \mathbf{I}(t')) = \mu(t)p(\mathbf{I}(t), \mathbf{I}(t'))$. This Markov chain $\mathbf{Y}$ corresponds to the routing chain as defined in Henderson *et al.* [13], [16]. Let $CI(\mathbf{x})$ be the equivalence class of $\mathbf{x} \in \mathcal{C}lT$, that is $CI(\mathbf{x}) = \{\mathbf{x}' | \mathbf{x}\ CI^*\ \mathbf{x}'\}$. The equivalence classes partition $\mathcal{C}lT$: each $\mathbf{x} \in \mathcal{C}lT$ belongs to exactly one equivalence class.

Let $\mathbf{x} \in \mathcal{C}lT$ with equivalence class $CI(\mathbf{x})$. Define $S(\mathbf{x}) \subset S$, the input bags corresponding to $CI(\mathbf{x})$, as

$$S(\mathbf{x}) = \{\mathbf{I}(t) | \exists\ \mathbf{x}' \in CI(\mathbf{x}) \text{ such that } x'_t > 0\}.$$

The following theorem shows that the partition of $\mathcal{C}lT$ into equivalence classes $\{CI(\mathbf{x})\}_{\mathbf{x} \in \mathcal{C}lT}$ induces a partition $\{S(\mathbf{x})\}_{\mathbf{x} \in \mathcal{C}lT}$ of S into irreducible sets of the Markov chain $\mathbf{Y}$ if and only if Assumption 3.2 is satisfied.

THEOREM 3.7 (STRUCTURAL CHARACTERISATION) *Assumption 3.2 is necessary and sufficient for the existence of an invariant measure for the traffic equations (2).*

Proof Observe that the state-independent traffic equations (2) are the global balance equations of $\mathbf{Y}$ at state space S. Therefore it is sufficient to prove that Assumption 3.2 is necessary and sufficient for the partition of S into irreducible sets $\{S(\mathbf{x})\}_{\mathbf{x} \in \mathcal{C}lT}$.

Let $\mathbf{x}, \mathbf{x}' \in \mathcal{C}lT$. If $\mathbf{x}' \in CI(\mathbf{x})$ then $S(\mathbf{x}') = S(\mathbf{x})$, since $CI(\mathbf{x}) = CI(\mathbf{x}')$. If $S(\mathbf{x}') \cap S(\mathbf{x}) \neq \emptyset$, then $\exists\ t \in T$ such that $\mathbf{I}(t) \in S(\mathbf{x}') \cap S(\mathbf{x})$ implying that $\exists\ \mathbf{x}'' \in CI(\mathbf{x})$ for which $\exists\ s \in T$ such that $x''_s > 0$ and $\mathbf{I}(s) = \mathbf{I}(t)$, and $\exists\ \mathbf{x}''' \in CI(\mathbf{x}')$ for which $\exists\ s'$ such that $x'''_{s'} > 0$ and $\mathbf{I}(s') = \mathbf{I}(t)$. Thus $CI(\mathbf{x}'') = CI(\mathbf{x}''')$ implying $CI(\mathbf{x}) = CI(\mathbf{x}')$, in turn implying that $S(\mathbf{x}') = S(\mathbf{x})$. This shows that $S(\mathbf{x}') = S(\mathbf{x})$ if $CI(\mathbf{x}') = CI(\mathbf{x})$, and $S(\mathbf{x}') \cap S(\mathbf{x}) = \emptyset$ if $CI(\mathbf{x}') \cap CI(\mathbf{x}) = \emptyset$.

Assumption 3.2 implies that for all $t \in T$, $\exists\ \mathbf{x} \in \mathcal{C}lT$ such that $t \in \|\mathbf{x}\|$, i.e. $\exists\ S(\mathbf{x})$ such that $\mathbf{I}(t) \in S(\mathbf{x})$. As a consequence $\{S(\mathbf{x})\}_{\mathbf{x} \in \mathcal{C}lT}$ forms a partition of S.

Let $\mathbf{I}(t), \mathbf{I}(t') \in S(\mathbf{x})$. Then $\exists\ \mathbf{x}', \mathbf{x}'' \in CI(\mathbf{x})$ for which $\exists\ s, s' \in T$ such that $x'_s > 0$ and $x''_{s'} > 0$, and $\mathbf{I}(s) = \mathbf{I}(t)$ and $\mathbf{I}(s') = \mathbf{I}(t)$, but also $\mathbf{x}' CI^* \mathbf{x}''$. Thus $\exists\ \sigma$, firing-sequence, such that $\mathbf{I}(t)[\sigma > \mathbf{I}(t')$. Let $\mathbf{I}(t) \in S(\mathbf{x})$, $\mathbf{I}(t') \in S(\mathbf{x}')$, $S(\mathbf{x}) \cap S(\mathbf{x}') = \emptyset$. Assume $\exists\ \sigma$, firing sequence, such that $\mathbf{I}(t)[\sigma > \mathbf{I}(t')$ then $\mathbf{x}' \in CI(\mathbf{x})$ implying that $S(\mathbf{x}) = S(\mathbf{x}')$. As a consequence $\{S(\mathbf{x})\}_{\mathbf{x} \in \mathcal{C}lT}$ forms a partition of S into irreducible sets. The Perron-Frobenius theorem (cf. Seneta [22]) implies that a positive solution exists to the marking independent traffic equations.

Conversely, assume that an invariant measure exists to the marking independent traffic equations. This immediately implies that for all $t \in T\ \exists\ t' \in T$

such that $\mathbf{O}(t) = \mathbf{I}(t')$. Furthermore, the existence of this invariant measure implies that S is partitioned in irreducible sets. Let V_i, $i = 1, \ldots, v$, denote the irreducible sets of $\mathbf{Y}$. Let $t \in T$ and i_0 such that $\mathbf{I}(t) \in V_{i_0}$. Since V_{i_0} is an irreducible set we have that for all $\mathbf{v} \in V_{i_0}$ $\exists\ \sigma, \sigma'$ such that $\mathbf{I}(t)[\sigma > \mathbf{v}$, and $\mathbf{v}[\sigma' > \mathbf{I}(t)$. Thus $\tilde{\sigma} = \sigma\sigma'$ is a closed support T-invariant. Similarly, from the irreducibility we may conclude that all T-invariants contained in V_{i_0} have closed support. From Memmi and Roucairol [19] we obtain that each support of an invariant can be decomposed into a union of minimal supports which implies that t is covered by a minimal closed support T-invariant. □

REMARK 3.8 (STRUCTURAL CHARACTERISATION) In the literature, one usually assumes that a solution for the traffic equations exists, and necessary conditions are derived from this assumption (e.g., Henderson *et al.* [13]). Theorem 3.7 provides a *necessary and sufficient* structural condition for the existence of a solution of the traffic equations, only. We will now illustrate the difference between Assumption 3.2 and the conditions of Henderson *et al.* [13], [16] that are *necessary* for the existence of a solution for the traffic equations. This also shows that Assumption 3.2 is a *new condition* for the characterisation of product form results.

Henderson *et al.* [13] introduce the following necessary condition for the existence of a solution for the traffic equations (Corollary 1): *for all* $\mathbf{g} \in \mathcal{R}(T) = \bigcup_{t \in T}\{\mathbf{I}(t) \cup \mathbf{O}(t)\}$ *there exist* $t, s \in T$ *such that* $\mathbf{g} = \mathbf{I}(t)$, $\mathbf{g} = \mathbf{O}(s)$, *that is* $\mathcal{R}(T)$ *is a closed set.* Obviously, Assumption 3.2 implies this condition, since Assumption 3.2 not only assumes that such $t, s \in T$ exist, but also that t, s are elements of the support of a single minimal closed support T-invariant. The reversed statement is not true, as is shown in the following example taken from Coleman [6], where the example is given to illustrate that the condition of Corollary 1 from Henderson *et al.* [13] is not sufficient for the existence of a solution for the traffic equations.

Consider the Petri net depicted in Figure 1. From the incidence matrix

$$A = \begin{pmatrix} -1 & 0 & -1 & 1 & 1 \\ 1 & -1 & -1 & -1 & 1 \\ -1 & 1 & 1 & 1 & -1 \\ 1 & 0 & 1 & -1 & -1 \end{pmatrix}$$

we obtain that this net has 3 minimal support T-invariants: $\mathbf{x}^1 = (10010)$, $\mathbf{x}^2 = (00101)$, $\mathbf{x}^3 = (12001)$, of which $\mathbf{x}^1$ and $\mathbf{x}^2$ have closed support, but $\mathbf{x}^3$ does not have closed support. (This can be seen from Theorem 3.3, or from the definition of closed sets.) Since transition t_2 is contained in $\|\mathbf{x}^3\|$ only, t_2 cannot be covered by a minimal closed support T-invariant, which contradicts Assumption 3.2. In contrast, the condition of Corollary 1 from Henderson *et al.* [13] is satisfied, also for transition t_2.

The state space of the routing chain is

$$S = \{\mathbf{I}(t_1), \mathbf{I}(t_2), \mathbf{I}(t_4), \mathbf{I}(t_5)\}, \quad (\mathbf{I}(t_2) = \mathbf{I}(t_3)),$$

and the solution of the traffic equations (2) is (up to a multiplicative constant)

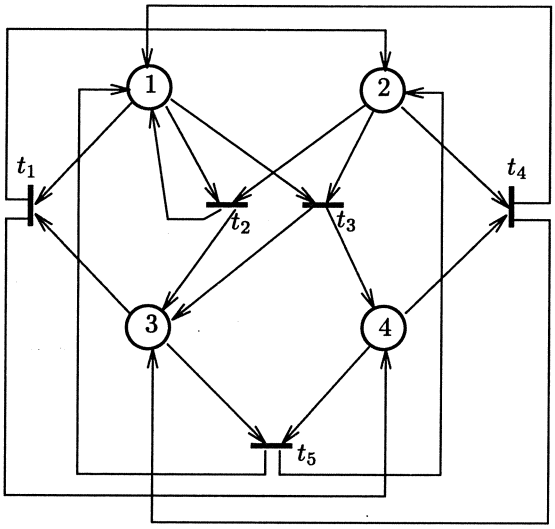


FIGURE 1. Petri net violating Assumption 3.2

$$y(\mathbf{I}(t_1)) = 1/\mu_1,\ y(\mathbf{I}(t_4)) = 1/\mu_4,\ y(\mathbf{I}(t_2)) = 0,\ y(\mathbf{I}(t_3)) = 0,\ y(\mathbf{I}(t_5)) = 0,$$

which shows that the condition of Corollary 1 from Henderson *et al.* [13] is not sufficient for the existence of a *positive* solution of the traffic equations. □

We are now able to present a first product form theorem for stochastic Petri nets. This theorem is formulated by analogy with similar results for batch routing queueing networks, and shows the similarity between stochastic Petri nets and batch routing queueing networks at the Markovian level.

THEOREM 3.9 *Assume that an invariant measure y exists to the marking independent traffic equations (2), and a function $\pi_y : \mathcal{M}(\mathbf{m}_0) \to \mathbb{R}^+$ such that for all $\mathbf{n} + I(t) \in \mathcal{M}(\mathbf{m}_0)$, $t, s \in T$ with $p(\mathbf{I}(t), \mathbf{I}(s)) > 0$,*

$$\frac{\pi_y(\mathbf{n} + \mathbf{I}(t))}{\pi_y(\mathbf{n} + \mathbf{I}(s))} = \frac{y(\mathbf{I}(t))}{y(\mathbf{I}(s))}. \tag{3}$$

Then $\pi_y(\mathbf{m})$, $\mathbf{m} \in \mathcal{M}(\mathbf{m}_0)$, is an invariant measure of the Markov chain $\mathbf{Y}$ describing the stochastic Petri net. If $B^{-1} = \sum_{\mathbf{m} \in \mathcal{M}(\mathbf{m}_0)} \pi_y(\mathbf{m}) < \infty$, then $\pi(\mathbf{m}) = B\pi_y(\mathbf{m})$, $\mathbf{m} \in \mathcal{M}(\mathbf{m}_0)$, is an equilibrium distribution of the Markov chain describing the stochastic Petri net.

The proof of Theorem 3.9 can be found in the literature (Boucherie and van Dijk [4], Henderson and Taylor [16]). The key-idea of Theorem 3.9 is that the marking independent solution $y(\cdot)$ of the traffic equations is translated into a marking dependent solution with the same properties. This is reflected in Condition (3). This establishes the *product form* nature of the equilibrium distribution.

Note that Condition (3) is a condition on y and *not* on the structure of the Petri net. Furthermore, as is shown in section 4.2, if a solution $y(\cdot)$ of the traffic equations is found, a function $\pi_y(\cdot)$ satisfying (3) cannot always be found without additional assumptions on the Petri net. We will now provide a structural characterisation of the Petri net guaranteeing (3). The rank condition is taken from Coleman *et al.* [7]. The result of this characterisation is that condition (3) is satisfied with a function π_y that is a product over the places of the Petri net.

THEOREM 3.10 *Assume that all transitions are covered by minimal closed support T-invariants. Then, with y the invariant measure for the traffic equations, π_y satisfying (3) has the form*

$$\pi_y(\mathbf{m}) = \prod_{i=1}^{N} c_i(y)^{\mathbf{m}(i)} \tag{4}$$

if and only if

$$Rank(A) = Rank([A|\mathbf{C}(y)]), \tag{5}$$

where $[A|\mathbf{C}(y)]$ is the matrix A augmented with the row $\mathbf{C}(y)$, defined as

$$\mathbf{C}(y)_j = \log\left[y(\mathbf{I}(t_j))/y(\mathbf{O}(t_j))\right], \quad j = 1, \ldots, M.$$

In this case the N-vector $\mathbf{c}(y) = (\log c_i(y),\ i = 1, \ldots, N)$ satisfies the matrix equation

$$\mathbf{c}(y)A + \mathbf{C}(y) = 0. \tag{6}$$

Observe that the solution y for the state-independent traffic equations is defined up to multiplicative factors at the irreducible sets of the routing chain $\mathbf{Y}$ at state space S only. This cannot give rise to problems in the above theorem, since we only use the ratios $y(\mathbf{I}(t))/y(\mathbf{I}(s))$, where $\mathbf{I}(t)$ and $\mathbf{I}(s)$ are in the same irreducible set of $\mathbf{Y}$, in the definition of $\mathbf{C}(y)$. This quotient is unique at each irreducible set, and therefore $\mathbf{C}(y)$ is uniquely determined.

Theorem 3.10 and its proof are taken from Coleman *et al.* [7]. This theorem characterises product forms for stochastic Petri nets based on the incidence matrix. The product form (4) is of the Jackson-type since it is a product over the places similar to the result of Jackson [17]. Note that the Petri nets are substantially more complex than Jackson networks.

Observe that Theorem 3.10 states that a product form solution (4) exists if and only if the invariant measure $y(\cdot)$ for the traffic equations is such that $\mathbf{C}(y)$ is orthogonal to the right null space of A containing all T-invariants. The product form distribution (4) contains one term for each token in the Petri net. Therefore, the only dependence between tokens lies in the normalising constant.

REMARK 3.11 (GENERALISATIONS) The results of this section can immediately be generalised to also include marking dependent firing rates

$$q(\mathbf{I}(t), \mathbf{O}(t); \mathbf{m} - \mathbf{I}(t)) = \mu(t) \frac{\psi(\mathbf{m} - \mathbf{I}(t))}{\phi(\mathbf{m})} p(\mathbf{I}(t), \mathbf{O}(t)),$$

where $\psi(\mathbf{m} - \mathbf{I}(t))/\phi(\mathbf{m})$ is the marking dependent firing rate. This does not affect the analysis as can be seen from the literature on batch routing queueing networks (cf. Boucherie and van Dijk [4]: $\mathbf{I}(t)$ and $\mathbf{O}(t)$ correspond to the batches of departing and arriving customers, $\mu(t)\psi(\mathbf{m} - \mathbf{I}(t))/\phi(\mathbf{m})$ is the service rate, and $p(\mathbf{I}(t), \mathbf{O}(t))$ is the routing probability for the customers in the batch). The equilibrium distribution becomes

$$\pi(\mathbf{m}) = B\phi(\mathbf{m})\pi_y(\mathbf{m}).$$

The inclusion of a marking dependent part in the firing rates allows for more general Petri nets. The structural analysis based on $p(\mathbf{I}(t), \mathbf{O}(t))$ is not affected, but some marking dependent properties can be modelled using ψ. Furthermore, $p(\mathbf{I}(t), \mathbf{O}(t))$ can be generalised to a marking dependent function $p(\mathbf{I}(t), \mathbf{O}(t); \mathbf{m} - \mathbf{I}(t))$, which allows us to introduce inhibitor arcs in the Petri net formalism. The Petri nets obtained via these two generalisations cannot be completely characterised at the structural level: some of the transitions that are enabled in the net with firing rates $\mu(t)p(\mathbf{I}(t), \mathbf{O}(t))$ can be excluded in a marking dependent way. Some results in this direction can be found in Boucherie and Sereno [5]. □

4 Examples

In this section we present some examples illustrating the structural characterisation presented above. First, in example 4.1 we present the product form results obtained by Lazar and Robertazzi [18]. In example 4.2 we present some examples of Petri nets that are covered by closed support T-invariants, but with different behaviour: a net that always has a product form equilibrium distribution, a net that sometimes has such a distribution, and a net that does not have an equilibrium distribution at all. This shows that closed support T-invariants can be rather complex, and illustrates the theoretical results of section 3.

4.1 The dual processor system

The Petri nets discussed by Lazar and Robertazzi [18] are of the form presented here. We will illustrate the framework of Lazar and Robertazzi with an example.

Consider the dual processor system. It consists of two processors sharing a single memory. The processors may refer to the shared memory through a bus. A processor is allowed to work only if the bus is available (!), hence conflicts between the processors occur as only one of the processors may utilize the bus. The assumption that the processors are allowed to work only if the bus is available is necessary to obtain a product form equilibrium distribution, and is reflected in the assumption of Robertazzi and Lazar that a task sequence is only allowed to proceed if there is a non-zero probability that it can return to its current state without the need for a state change in other task sequences.

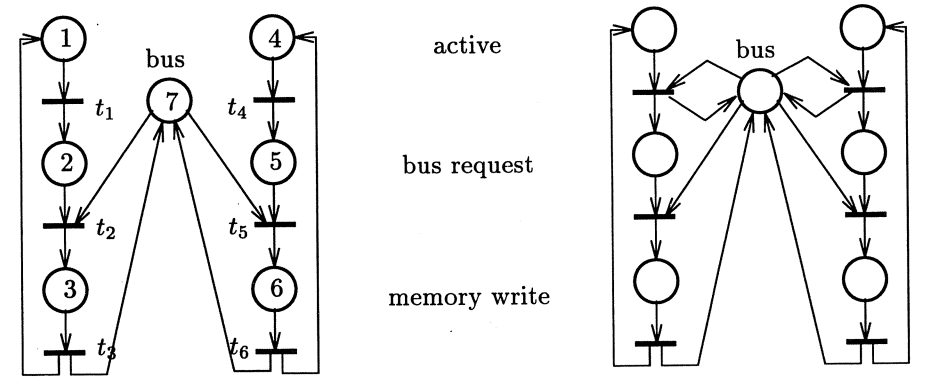


FIGURE 2.

The practical consequence of this assumption is that arcs are added in the Petri net of Figure 2a representing the dual processor system without modifications. This results in the Petri net of Figure 2b representing the dual processor system in which processors can work only when the bus is available.

The Petri nets of Figure 2 have two T-invariants $\mathbf{x}^1 = (111000)$, $\mathbf{x}^2 = (000111)$. As a consequence of the extra arcs which are added because we have assumed that a processor is allowed to work only if the bus is available, both T-invariants for the Petri net of Figure 2b are minimal closed support T-invariants. Note that the Petri net without the extra arcs has the same two minimal support T-invariants. This is an immediate consequence of the fact that the extra arcs do not contribute to the incidence matrix A, which shows that Assumption 3.2 cannot be verified on the basis of the incidence matrix A only, but needs to be verified directly from the input and output functions $I(\cdot,\cdot)$ and $O(\cdot,\cdot)$.

The transition rates of the Petri net are of the form (1):

$$q(\mathbf{I}(t_i), \mathbf{O}(t_i); \mathbf{m} - \mathbf{I}(t_i)) = \mu(t_i),$$

for $i = 1, \ldots, 6$, such that $\mathbf{m} - \mathbf{I}(t_i) \in \mathbb{N}_0^N$. In Lazar and Robertazzi [18] initially one token is present at places 1, 4 and 7. The equilibrium distribution is

$$\pi(\mathbf{m}) = B \prod_{i=1}^{6} (1/\mu(t_i))^{m_i}, \quad \mathbf{m} \in \mathcal{M}(\mathbf{m}_0),$$

where the reachability set $\mathcal{M}(\mathbf{m}_0)$ is

$$\mathcal{M}(\mathbf{m}_0) = \mathcal{M}(1001001) = \{(1001001), (0101001), (0011000), (1000101), (0100101), (0010100), (1000101), (1000101)\}.$$

From Theorem 3.9 we obtain that except for the normalisation constant B, the equilibrium distribution has the same form if the assumption of safeness (at most one token in each place) made by Lazar and Robertazzi [18] is removed. The only difference is the reachability set $\mathcal{M}(\mathbf{m}_0)$. This result shows the power of the use of T-invariants in the analysis of Petri nets: the form of the equilibrium distribution is completely determined by the T-invariants, regardless of the shape of the reachability set.

4.2 Closed support T-invariants

This example considers three stochastic Petri nets that are covered by closed support T-invariants, but with completely different behaviour. The Petri net of Figure 3a has a product form equilibrium distribution, the net of Figure 3b has a product form equilibrium distribution for a specific choice of the firing rates (related to conflicting T-invariants), and the net of Figure 3c may not possess an equilibrium distribution (due to a possibly unbounded number of tokens).

Consider the Petri net depicted in Figure 3a. From the incidence matrix

$$A = \begin{pmatrix} -1 & -1 & 1 & 0 & 0 \\ 1 & 0 & -1 & 1 & 0 \\ 2 & 1 & -2 & 2 & -1 \\ 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix},$$

we obtain that this net has two minimal support T-invariants $\mathbf{x}^1 = (10100)$, $\mathbf{x}^2 = (01111)$, which are both minimal closed support T-invariants, and two minimal support P-invariants $\mathbf{y}^1 = (11011)$, $\mathbf{y}^2 = (20112)$. Since the T-invariants share $\mathbf{I}(t_1)$ they are in common input bag relation, which implies that the routing chain has one irreducible set: $S = \{\mathbf{I}(t_1), \mathbf{I}(t_3), \mathbf{I}(t_4), \mathbf{I}(t_5)\}$ $(\mathbf{I}(t_1) = \mathbf{I}(t_2))$.

Denote $\mu(t_{12}) = \mu(t_1) + \mu(t_2)$, $b = \mu(t_2)/\mu(t_{12})$, the probability that transition t_2 fires before transition t_1 when transitions t_1 and t_2 are enabled. The solution of the traffic equations is (up to normalisation)

$$y(\mathbf{I}(t_1))\mu(t_{12}) = y(\mathbf{I}(t_3))\mu(t_3) = 1, \quad y(\mathbf{I}(t_4))\mu(t_4) = y(\mathbf{I}(t_5))\mu(t_5) = b.$$

The solution π_y to (3) is not immediately obvious from these relations, therefore we apply Theorem 3.10 to derive this solution. The vector $\mathbf{C}(y)$ can be obtained from the solution of the traffic equations:

$$\mathbf{C}(f) = \left(\log\left[\tfrac{\mu(t_3)}{\mu(t_{12})}\right], \log\left[\tfrac{\mu(t_5)}{b\mu(t_{12})}\right], \log\left[\tfrac{\mu(t_{12})}{\mu(t_3)}\right], \log\left[\tfrac{b\mu(t_3)}{\mu(t_4)}\right], \log\left[\tfrac{\mu(t_4)}{\mu(t_5)}\right]\right).$$

It can easily be verified that $Rank(A) = Rank(A|\mathbf{C}(y))$ without any conditions on the firing rates. The solution $\mathbf{c}(y)$ of the system of equations (6) is (we have set $c_1(y) = c_3(y) = 1$ as normalisation)

$$c_1(y) = 1,\ c_2(y) = \frac{\mu(t_{12})}{\mu(t_3)},\ c_3(y) = 1,\ c_4(y) = \frac{b\mu(t_{12})}{\mu(t_5)},\ c_5(y) = \frac{b\mu(t_{12})}{\mu(t_4)}$$

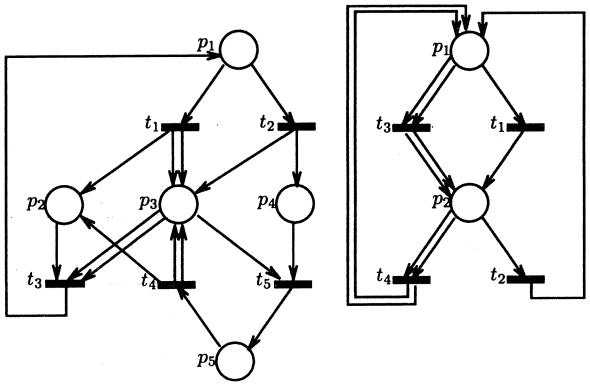


FIGURE 3. a.

FIGURE 3: b.

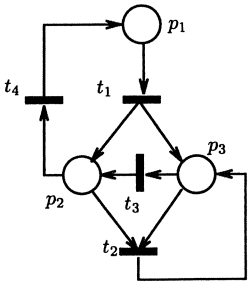


FIGURE 3: c.

is a solution to (6), and the equilibrium distribution is (cf. Coleman *et al.* [7])

$$\pi_y(\mathbf{m}) = \left(\frac{\mu(t_{12})}{\mu(t_3)}\right)^{m(2)} \left(\frac{b\mu(t_{12})}{\mu(t_5)}\right)^{m(4)} \left(\frac{b\mu(t_{12})}{\mu(t_4)}\right)^{m(5)}$$

is an invariant measure for the Petri net at reachability set

$$\mathcal{M}(\mathbf{m}_0) = \{\mathbf{m} : \mathbf{y}^1 \bullet (\mathbf{m} - \mathbf{m}_0) = 0,\ \mathbf{y}^2 \bullet (\mathbf{m} - \mathbf{m}_0) = 0\},$$

where $\bullet$ denotes the inner product of the two vectors.

Consider the Petri net depicted in Figure 3b. This Petri net has incidence matrix

$$A = \begin{pmatrix} -1 & 1 & -2 & 2 \\ 1 & -1 & 2 & -2 \end{pmatrix}.$$

Observe that each transition is covered by the minimal closed support T-invariants $\mathbf{x}^1 = (1100)$, $\mathbf{x}^2 = (0011)$, but that $\mathbf{x}^3 = (2001)$, and $\mathbf{x}^4 = (0210)$, are also minimal support T-invariants that do not have closed support.

The routing chain has two irreducible sets $S(\mathbf{x}^1) = \{\mathbf{I}(t_1), \mathbf{I}(t_2)\}$, and $S(\mathbf{x}^2) = \{\mathbf{I}(t_3), \mathbf{I}(t_4)\}$. Theorem 3.9 implies that the traffic equations have a positive solution. This solution is

$$\frac{y^1(I(t_2))}{y^1(I(t_1))} = \frac{\mu(t_1)}{\mu(t_2)}, \quad \frac{y^2(I(t_4))}{y^2(I(t_3))} = \frac{\mu(t_3)}{\mu(t_4)},$$

with corresponding vector $\mathbf{C}(y)$

$$\mathbf{C}(y) = \left(\log\left[\frac{\mu(t_2)}{\mu(t_1)}\right], \log\left[\frac{\mu(t_1)}{\mu(t_2)}\right], \log\left[\frac{\mu(t_4)}{\mu(t_3)}\right], \log\left[\frac{\mu(t_3)}{\mu(t_4)}\right]\right).$$

The matrix $[A|\mathbf{C}(y)]$ is

$$[A|\mathbf{C}(y)] = \begin{pmatrix} -1 & 1 & -2 & 2 \\ 1 & -1 & 2 & -2 \\ C_1 & C_2 & C_3 & C_4 \end{pmatrix},$$

and $Rank([A|\mathbf{C}(y)]) = Rank(A) = 1$ if and only if $C_1 + C_2 = 0$, $2C_1 - C_3 = 0$, $2C_1 + C_4 = 0$, that is if and only if

$$\left(\frac{\mu(t_2)}{\mu(t_1)}\right)^2 = \frac{\mu(t_4)}{\mu(t_3)}. \tag{7}$$

If this is the case, the Petri net has an equilibrium distribution

$$\pi(\mathbf{m}) = B\left(\frac{\mu(t_2)}{\mu(t_1)}\right)^{\mathbf{m}(1)},$$

at reachability set

$$\mathcal{M}(\mathbf{m}_0) = \{\mathbf{m} : \mathbf{m}(1) + \mathbf{m}(2) = \mathbf{m}_0(1) + \mathbf{m}_0(2)\}.$$

This example provides an interpretation and explanation of the rank condition (5) of Theorem 3.10. As can be seen from Figure 3b, for two tokens to move from place 1 to place 2 we have two possibilities. In the first case (via t_1) the tokens jump one after the other, in the second case (via t_3) the tokens jump simultaneously. The probability flow for these two possibilities must be the same. This is reflected in the condition (7) on the firing rates: two transitions with rate $\mu(t_1)$ must be proportional to one transition at rate $\mu(t_3)$.

Finally, consider the Petri net of Figure 3c. The Petri net has one T-invariant $\mathbf{x} = (1111)$ covering all transitions, and $\mathbf{x}$ has closed support. From Theorem 3.7 we obtain that the traffic equations have a positive solution. This solution is (up to a multiplicative constant)

$$y(\mathbf{I}(t_1)) = 1/\mu(t_1), \quad y(\mathbf{I}(t_2)) = 1/\mu(t_2), \quad y(\mathbf{I}(t_3)) = 1/\mu(t_3),$$

$$y(\mathbf{I}(t_4)) = 1/\mu(t_4),$$

and the Petri net has an invariant measure

$$m(\mathbf{m}) = \left(\frac{\mu(t_3)\mu(t_4)}{\mu(t_1)\mu(t_2)}\right)^{\mathbf{m}(1)} \left(\frac{\mu(t_3)}{\mu(t_2)}\right)^{\mathbf{m}(2)} \left(\frac{\mu(t_4)}{\mu(t_2)}\right)^{\mathbf{m}(3)}.$$

From Figure 3c we can see that the number of tokens in the net is unbounded (repetitive firing of transitions t_1 and t_4 increases the number of tokens by 1), but that for every marking a firing sequence to $\mathbf{m}_0 = (100)$ exists. If $\mu(t_3)\mu(t_4) < \mu(t_1)\mu(t_2)$, $\mu(t_3) < \mu(t_2)$, $\mu(t_4) < \mu(t_2)$ the Petri net has an equilibrium distribution

$$\pi(\mathbf{m}) = Bm(\mathbf{m}), \quad \mathbf{m} \in \mathcal{M}(\mathbf{m}_0) = \mathbb{N}_0^3 \setminus \{0\}.$$

REFERENCES

1. Ajmone Marsan, M., Balbo, G., Bobbio, A., Chiola, G., Conte, G. and Cumani, A. (1989) The effect of execution policies on the semantics and analysis of stochastic Petri nets, *IEEE Transactions on Software Engineering* **15**, 832-846.
2. Baskett, F., Chandy, K.M., Muntz, R.R. and Palacios, F.G. (1975) Open, closed and mixed networks of queues with different classes of customers, *Journal of the ACM* **22**, 248-260.
3. Boucherie, R.J. (1993) A characterisation of independence for competing Markov chains with applications to stochastic Petri nets, *Proceedings of the 5th International Workshop on Petri Nets and Performance Models (PNPM93), Toulouse, France, October 19-22, 1993*, 117-126.
4. Boucherie, R.J. and Van Dijk, N.M. (1991) Product forms for queueing networks with state dependent multiple job transitions, *Advances in Applied Probability* **23**, 152-187.
5. Boucherie, R.J. and Sereno, M. (1994) Product forms for stochastic Petri nets, *in preparation.*
6. Coleman, J.L. (1993) Ph.D. thesis, University of Adelaide, in preparation.
7. Coleman, J.L., Henderson, W. and Taylor, P.G. (1992) Product form equilibrium distributions and an algorithm for classes of batch movement queueing networks and stochastic Petri nets, *Research Report, University of Adelaide.*
8. Donatelli, S. and Sereno, M. (1992) On the product form solution for stochastic Petri nets, *Proceedings of the 13th international conference on application and theory of Petri nets, Sheffield, UK, 1992*, 154-172.
9. Florin, G. and Natkin, S. (1991) Generalization of queueing network product form solutions to stochastic Petri nets, *IEEE Transactions on Software Engineering* **17**, 99-107.
10. Frosch, D. (1992) Product form solutions for closed synchronized systems of stochastic sequential processes, Forschungsbericht Nr. 92-13, Universität Trier, Mathematik/Informatik.
11. Frosch, D. and Natarajan, K. (1992) Product form solutions for closed synchronized systems of stochastic sequential processes, *Proceedings of 1992 International Computer Symposium, December 13-15, Taichung, Taiwan*, 392-402.
12. Frosch-Wilke, D. (1993) Exact performance analysis of a class of product form stochastic Petri nets, *Proceedings of the 1993 UK Performance Engineering Workshop for Computer and Telecommunication Systems, Loughborough, UK, July 1993.*
13. Henderson, W., Lucic, D. and Taylor, P.G. (1989) A net level performance analysis of stochastic Petri nets, *Journal of the Australian Mathematical Society Series B* **31**, 176-187.
14. Henderson, W. and Taylor, P.G. (1989) Aggregation methods in exact performance analysis of stochastic Petri nets, *Proceedings of PNPM'89, Kyoto, Japan, December 1989*, pp. 12-18.
15. Henderson, W. and Taylor, P.G. (1990) Open networks of queues with batch

arrivals and batch services, *Queueing Systems* **6**, 71-88.
16. Henderson, W. and Taylor, P.G. (1991) Embedded processes in stochastic Petri nets, *IEEE Transactions on Software Engineering* **17**, 108-116.
17. Jackson, J.R. (1957) Networks of waiting lines, *Operations Research* **5**, 518-521.
18. Lazar, A.A. and Robertazzi, T.G. (1987) Markovian Petri net protocols with product form solution, *In: Proceedings of IEEE Infocom'87, San Francisco, CA, March 1987*, pp. 1054-1062. Also: *Performance Evaluation* **12**, 67-77 (1991).
19. Memmi, G. and Roucairol, G. (1979) Linear algebra in net theory, *In: Net theory and applications, Proceedings of the Advanced Course on General Net Theory of Processes and Systems, Hamburg, 1979, Lecture Notes in Computer Science 84*, pp. 213-223.
20. Molloy, M.K. (1982) Performance analysis using stochastic Petri nets, *IEEE Transactions on Computers*, **31**, 913-917.
21. Murata, T. (1989) Petri nets: properties, analysis and applications, *Proceedings of the IEEE* **77**, 541-580.
22. Seneta, E. (1981) *Non-negative matrices and Markov chains.* Springer-Verlag.
23. Serfozo, R.F. (1989) Markovian network processes: congestion dependent routing and processing, *Queueing Systems* **5**, 5-36.

Computational Algorithms for Product Form Solution Stochastic Petri Nets*

Matteo Sereno and Gianfranco Balbo
Dipartimento di Informatica,
Università di Torino,
Corso Svizzera 185,
10149 Torino, Italy

The combinatorial explosion of the state space of Stochastic Petri Nets (SPN) is a well known problem that inhibits the exact solution of large SPNs and thus a broad use of this kind of formalism as a modelling tool. In this paper we show that the steady state probability distribution of SPNs with product form solution can be efficiently computed using an algorithm whose space and time complexities are polynomial in the number of places and in the number of tokens in the initial marking of the SPN.
Basic to the derivation of such an algorithm is a product form solution criterion proposed by J. Coleman, W. Henderson and P.G. Taylor. The algorithm relies on the derivation of a recursive expression of the normalization constant that is a generalization of that derived by J.P. Buzen for multiple class product form queueing networks with load independent service centers.

1 Introduction

Stochastic Petri Nets (SPN) are a powerful tool for modelling and evaluating the performance of systems involving concurrency, non determinism and synchronization, such as parallel and distributed systems and communication networks.

SPNs have been proven to be equivalent to Continuous Time Markov Chains and their steady state analysis can thus be expressed as the solution of a system of equilibrium equations, one for each possible marking of their state space. The major problem in the computation of performance measures using SPNs is thus the size of the reachability set of these models that increases exponentially both with the number of tokens in the initial marking and with the number of places in the net. As a consequence, except for special classes of models, the dimension of this reachability set and the time complexity of the solution procedure preclude the exact numerical evaluation of many interesting models.

*Supported by the European Grant BRA-QMIPS of CEC DG XIII.

Recently, certain classes of SPNs have been discovered [9, 10, 11, 13] that are characterized by a steady state probability distribution of their markings that can be factorized, yielding a so called *Product Form Solution* (PFS). In particular, for the class of SPNs identified with the criterion proposed by Coleman, Henderson, Lucic and Taylor [9, 10, 11, 6], the factorization contains as many terms as there are places in the net. In this case the PFS resembles that of a class of Queueing Networks (QN) [12, 8, 2] for which efficient computational algorithms have been derived, making QN models truly effective for the performance evaluation of many real systems [4, 16, 15, 3].

In this paper we consider this last class of SPN and we derive an efficient algorithm for the computation of the PFS that has polynomial time and space complexities and that recalls the convolution algorithm derived by Buzen [4] for PFS Queueing Networks.

The balance of the paper is the following : Section 2 presents the necessary notation; in Section 3 we briefly introduce the criterion used to identify the SPN with PFS that we consider, and we recall the basic results that characterize this class of models. Section 4 contains the actual contribution of this paper, presenting a set of recursive equations that yield a convolution algorithm to compute the normalization constant and several performance measures for this class of SPN. Section 5 presents an example of application of this algorithm for the evaluation of an interesting model. Finally Section 6 concludes the paper outlining possible future works on this topic.

2 Notation and Basic Definitions

In this section we introduce the notation that will be used throughout the whole paper and we list the basic definitions that characterize the Petri Net formalism.

Definition 1 **[Stochastic Petri Net]** *A continuous time Stochastic Petri Net can be defined as a six-tuple:*

$$(\mathcal{P}, \mathcal{T}, I, O, \mathcal{R}, \mathbf{m_0}),$$

where:

- $\mathcal{P} = \{p_1, p_2, \ldots, p_P\}$ *is a set of places;*
- $\mathcal{T} = \{t_1, t_2, \ldots, t_T\}$ *is a set of transitions;*
- $\mathcal{P} \cap \mathcal{T} = \emptyset \ \wedge \ \mathcal{P} \cup \mathcal{T} \neq \emptyset$;
- $I, O : \mathcal{T} \times \mathcal{P} \rightarrow I\!N$ *are the input and output functions, identifying the arcs that connect places to transitions and transitions to places;*
- $\mathcal{R} = \{\mu(t_1), \mu(t_2), \ldots, \mu(t_T)\}$ *is a set of firing rates for the exponentially distributed transition firing times;*
- $\mathbf{m_0}$ *is the initial marking.*

From the functions $I(.,.)$ and $O(.,.)$ we derive the vectors $\mathbf{I}(t) = [I_1(t), I_2(t), \ldots, I_P(t)]$ and $\mathbf{O}(t) = [O_1(t), O_2(t), \ldots, O_P(t)]$, $\forall\, t \in \mathcal{T}$; where $I_i(t) = I(t, p_i)$ and $O_j(t) = O(t, p_j)$ $\forall\, p_i, p_j \in \mathcal{P}$. The vectors $\mathbf{I}(t)$ and $\mathbf{O}(t)$ $\forall\, t \in \mathcal{T}$, are called *input* and *output bags* of transition t. The input bag $\mathbf{I}(t)$ of transition t gives the enabling condition of t, i.e., a transition t is enabled in a given marking $\mathbf{m}$ iff $\mathbf{m} \geq \mathbf{I}(t)$. Any transition t that is enabled in a marking $\mathbf{m}$ can fire producing a new marking $\mathbf{m'} = [m'(p_1), \ldots, m'(p_P)]$ where $m'(p_i) = m(p_i) - I_i(t) + O_i(t)$, with $1 \leq i \leq P$ (symbolically $\mathbf{m}[t > \mathbf{m'}$).

DEFINITION 2 [**Reachability Set**] *Given a SPN* $(\mathcal{P}, \mathcal{T}, I, O, \mathcal{R}, \mathbf{m_0})$, *the* reachability set, $[\mathbf{m_0}>$, *is defined as follows:*

- $\mathbf{m_0} \in [\mathbf{m_0}>$
- *if* $\mathbf{m_i} \in [\mathbf{m_0}>$ *and* $\exists\, t \in \mathcal{T} : \mathbf{m_i}[t> \mathbf{m_{i+1}}$ *then* $\mathbf{m_{i+1}} \in [\mathbf{m_0}>$.

DEFINITION 3 [**Incidence Matrix**] *The* incidence matrix $\mathbf{A}$ *is a* $T \times P$ *matrix. The entries* $A[t, p] = I(t, p) - O(t, p)$, *with* $t \in \mathcal{T}$ *and* $p \in \mathcal{P}$, *represent the change in the number of tokens at place* p *when transition* t *fires.*

DEFINITION 4 [**S-invariant**] *A* S-invariant $\mathbf{s}$ *is a non negative row vector over the places of the net such that* $\mathbf{A.s} = \mathbf{0}$. *The weighted sum of the number of tokens distributed over the places corresponding to non-zero entries of a S-invariant remains constant as the marking process evolves. An invariant* $\mathbf{s}$ *is called* minimal *if there is no other invariant* $\mathbf{s'}$ *such that* $\mathbf{s'} \leq \mathbf{s}$ *(component-wise inequality). All S-invariants can be obtained as linear combinations of a set of minimal S-invariants.* $\mathbf{S}$ *is the matrix whose rows are the minimal S-invariants.*

DEFINITION 5 [**T-invariant**] *A* T-invariant $\mathbf{x}$ *is a non negative vector over the transitions of the net such that* $\mathbf{A^T.x} = \mathbf{0}$. *An invariant* $\mathbf{x}$ *is called* minimal *if there is no other invariant* $\mathbf{x'}$ *such that* $\mathbf{x'} \leq \mathbf{x}$. *All T-invariants can be obtained as linear combinations of a set of minimal T-invariants.*

Invariants characterize the structural properties of a SPN: T-invariants represent sequences of transitions whose firing may bring the net back to its initial state; S-invariants identify bounded SPNs (i.e., SPNs such that the maximum number of tokens in their places in any reachable marking is finite) when they cover all the places of the nets. There are results showing that any *live* and *bounded* SPN is covered by T-invariants [17]. In the following we restrict ourselves to nets that are live and bounded.

3 PRODUCT FORM SOLUTION FOR STOCHASTIC PETRI NETS

The aim of this section is the definition of the PFS framework in which the algorithms for the computation of the solutions will be derived. The PFS for SPN criterion considered here is that proposed by Coleman, Henderson, Lucic and Taylor [9, 10, 11, 6]. Since our interest is focused on the algorithms developed from the PFS concepts (the algorithms will be presented in Section 4), in this section we only recall the results concerning these models; interested readers can find the details of the derivations in the cited references.

3.1 The preliminary transformation

We can assume that there is a one to one correspondence between input bags and transitions. Any SPN that does not satisfy this requirement can be modified according to the following definition:

DEFINITION 6 **[Input bag transformation (I-bag)]**
Given a SPN $(\mathcal{P}, \mathcal{T}, I, O, \mathcal{R}, \mathbf{m_0})$, *if there are two transitions* $t', t'' \in \mathcal{T}$ *such that* $\mathbf{I}(t') = \mathbf{I}(t'')$ *with firing rates* $\mu(t')$ *and* $\mu(t'')$, *we amalgamate the transitions* t' *and* t'' *into a single transition* t *with firing rate* $\mu(t) = \mu(t') + \mu(t'')$. *If* $\mathbf{O}(t')$ *and* $\mathbf{O}(t'')$ *are the output bags of* t' *and* t'', *transition* t *has two output bags:* $\mathbf{O}_1(t) = \mathbf{O}(t')$ *and* $\mathbf{O}_2(t) = \mathbf{O}(t'')$. *If* t *fires in a marking* $\mathbf{m}$, *the next marking is* $\mathbf{m}' = \mathbf{m} - \mathbf{I}(t) + \mathbf{O}_j(t)$ *with probability* $P(\mathbf{I}(t), \mathbf{O}_j(t))$ $(j = 1, 2)$, *where* $P(\mathbf{I}(t), \mathbf{O}_1(t)) = \frac{\mu(t')}{\mu(t')+\mu(t'')}$ *and* $P(\mathbf{I}(t), \mathbf{O}_2(t)) = \frac{\mu(t'')}{\mu(t')+\mu(t'')}$.

Obviously the I-bag transformation can be applied to any set of transitions sharing the same input bag. In this case the generalization of the previous definition is immediate. Denoting with B_t the number of output bags of transition t, we identify with $\mathbf{O}_j(t)$ the j-th output bag of transition t, with $j = 1, \ldots, B_t$. $\mathcal{B}$ is the set of all the output bags of a SPN, with $B = |\mathcal{B}| = \sum_{t \in \mathcal{T}} B_t$. Given a marking $\mathbf{m}$, we denote with $P(\mathbf{I}(t), \mathbf{O}_j(t))$ the probability that the next marking is $\mathbf{m}' = \mathbf{m} - \mathbf{I}(t) + \mathbf{O}_j(t)$.

For any $t \in \mathcal{T}$, if there exists an integer j, with $j = 1, \ldots, B_t$, such that $\mathbf{O}_j(t) = \mathbf{I}(s)$ for a certain transition $s \in \mathcal{T}$, we denote the transition s as $E_j(t)$ and we write $P(t, s) = P(\mathbf{I}(t), \mathbf{O}_j(t))$.

Remark: Every transition in the original net can be identified with the pair input/output bag, hence the original and transformed SPN have the same incidence matrix.

3.2 The routing process

The main feature of this approach for identifying SPNs with PFS is to consider the transitions of the SPN to be themselves states of a Markov Chain, which has been called *routing process* [9]. This interpretation is obtained by considering the input and output bags as states of a Markov chain and by finding a one to one correspondence between the states of this chain and the set of transitions of the SPN that holds under certain conditions. The conditions that a SPN has to satisfy in order for this correspondence to exist and that thus represent the characterization of the class of SPNs for which this PFS can be found, are resumed by the following definition.

DEFINITION 7 **[Structural Conditions]**

1. *No two transitions have the same input bag.*
2. *For each transition* $t \in \mathcal{T}$, $j = 1, \ldots, B_t$, $\mathbf{O}_j(t) = \mathbf{I}(s)$ *and for some transition* $s \in \mathcal{T}$, *we denote the transition* s *by* $E_j(t)$ *and we write* $P(t, s) = P(\mathbf{I}(t), \mathbf{O}_j(t))$.

3. *For every transition $s \in \mathcal{T}$ there must exist a transition $t \in \mathcal{T}$ and a $j = 1, \ldots, B_t$, such that $\mathbf{I}(s) = \mathbf{O}_j(t)$.*

These conditions allow to identify a discrete time Markov chain on the set of transitions of a SPN whose single step transition probabilities are given by $P(t, s)$.

DEFINITION 8 [**Routing Chain**] *The Markov chain with state space $\mathcal{T}$ and (one step) transition probabilities $P(t, s), t, s \in \mathcal{T}$ is called* routing chain *for the SPN. We denote the transition matrix of this Markov chain by* $\mathbf{P}$.

The set $\mathcal{D}$ of functions d from $\mathcal{T}$ to $[0, \infty)$ such that $d(t)\mu(t)$ is an invariant measure for the routing chain, plays an important role in this PFS criterion. This set is formally defined as

$$\mathcal{D} = \left\{ d(.),\ \mathcal{T} \to [0, \infty) :\ \mu(t)d(t) = \sum_{s \in \mathcal{T}} \mu(s)d(s)P(s, t),\ \forall\ t \in \mathcal{T} \right\}. \quad (1)$$

The authors of this PFS proposal showed that Definition 7 gives necessary conditions for the SPN to be such that the set $\mathcal{D}$ is not empty.

It is interesting to observe that the system of linear equations contained in (1) is homogeneous and hence admits (if the SPN satisfies the conditions of Definition 7) an infinite number of solutions. These solutions differ by a multiplicative constant.

3.3 The Product Form Solution

The equilibrium distribution for SPNs satisfying the conditions of Definition 7 is given by the following theorem proven by Henderson, Lucic, and Taylor in [9].

THEOREM 1 [**Product Form Solution**] *Assume that there exists a function $d(.) \in \mathcal{D}$ and a function $h(.) : [\mathbf{m_0}> \to R$ such that*

$$\frac{h(\mathbf{m} + \mathbf{I}(t))}{h(\mathbf{m} + \mathbf{I}(\mathbf{s}))} = \frac{d(t)}{d(s)} \qquad \forall\ s, t \in \mathcal{T}\ :\ P(s, t) > 0 \quad (2)$$

Then the equilibrium distribution of the SPN is given by

$$\pi(\mathbf{m}) = \frac{1}{G} \cdot h(\mathbf{m}) \qquad \mathbf{m} \in [\mathbf{m_0}>, \quad (3)$$

where G is a normalization constant. $\diamondsuit$

The function $h(\mathbf{m})$ represents the PFS of this type of SPNs. In order to obtain computationally efficient algorithms for its evaluation, it is convenient that the PFS contains as many terms as there are places or transitions in the SPN. In [6] the PFS has been found to be a product over the places of the SPN subject to the minimal S-invariants when the following additional condition holds.

Assume that a column vector $\mathbf{C}(d)$ is defined in the following manner [6] :

$$\mathbf{C}(d) = \begin{bmatrix} log\left(\frac{d(1)}{d(E_1(1))}\right) \\ \vdots \\ log\left(\frac{d(1)}{d(E_{B_1}(1))}\right) \\ \vdots \\ log\left(\frac{d(T)}{d(E_1(T))}\right) \\ \vdots \\ log\left(\frac{d(T)}{d(E_{B_T}(T))}\right) \end{bmatrix}, \tag{4}$$

for $i = 1, \ldots, B_t$, $t \in \mathcal{T}$ and $E_j(t)$ as defined in the conditions of Definition 7.

THEOREM 2 **[Form of $h(.)$; see [6]]** *Let $(\mathcal{P}, \mathcal{T}, I, O, \mathcal{R}, \mathbf{m_0})$ be a SPN such that $\mathcal{D}$ is non empty. The function $h(\mathbf{m})$ required to satisfy Theorem 1 is of the form*

$$h(\mathbf{m}) = \prod_{i \in \mathcal{P}} f_i(d)^{m(i)} \tag{5}$$

if and only if

$$Rank([\mathbf{A}]) = Rank([\mathbf{A} \mid \mathbf{C}(d)]), \tag{6}$$

where $[\mathbf{A} \mid \mathbf{C}(d)]$ is the matrix $\mathbf{A}$ augmented with the matrix $\mathbf{C}(d)$.
In this case $f_i(d), i \in \mathcal{P}$, satisfies the matrix equation

$$-\mathbf{A} \begin{bmatrix} log\,(f_1(d)) \\ \vdots \\ log\,(f_P(d)) \end{bmatrix} = \mathbf{C}(d). \tag{7}$$

$\Diamond$

4 The Normalization Constant Calculus

The straightforward computation of the normalization constant G that derives from definition (3)

$$G = \sum_{\mathbf{m} \in [\mathbf{m_0}>} h(\mathbf{m}) \tag{8}$$

is still exponentially complex and a calculus based on relationships among normalization constants computed for a smaller set of places of the net is needed in order to overcome this problem. In this section we show that indeed interesting recursive expressions can be found that yield a convolution algorithm that has polynomial complexity in terms of the number of places and of the initial marking of the net.

4.1 The Reachability Condition

The major problem in finding the performance indices of a SPN with product form equilibrium distribution is the computation of the normalization constant. To devise an algorithm for the computation of the PFS for a SPN we first restrict the class of nets that we are considering to that for which the following reachability condition holds.

Let $\mathbf{S}$ be the matrix whose rows are the minimal S-invariants.

DEFINITION 9 [**Reachability Condition**] *Given a SPN with initial marking* $\mathbf{m_0}$ *and the matrix* $\mathbf{S}$, *a necessary and sufficient condition for the reachability of any marking* $\mathbf{m}$ *is*

$$\mathbf{S}.\mathbf{m_0^T} = \mathbf{S}.\mathbf{m^T}. \tag{9}$$

With $\mathbf{S}.\mathbf{m_0^T}$ we denote the product of $\mathbf{S}$ by $\mathbf{m_0^T}$ that gives the initial distribution of tokens for each S-invariant. In the following the vector $\mathbf{K}$, such that $\mathbf{K} = \mathbf{S}.\mathbf{m_0^T}$, is called *load vector*, while the SPNs satisfying (9) are called *S-invariant reachable.*

Note that (9) is always necessary for the reachability of a certain marking $\mathbf{m}$; in this case we are also requiring that the condition be sufficient. There are several classes of SPNs which fall into this category. In particular, Equal Conflict systems, that naturally generalize the ordinary subclass of Free Choice systems, are S-invariant reachable in some cases [18].

Given a S-invariant reachable SPN $(\mathcal{P}, \mathcal{T}, I, O, \mathcal{R}, \mathbf{m_0})$ with load vector $\mathbf{K} = \mathbf{S}.\mathbf{m_0^T}$, the following notation represents an alternative way of denoting the reachability set:

$$\mathcal{E}(\mathbf{K}, \mathcal{P}) = \left\{\mathbf{m} : \mathbf{S}.\mathbf{m^T} = \mathbf{K}\right\}. \tag{10}$$

The reachability set of these SPNs can be partitioned according to several criteria. The first is that of classifying the markings according to the total number of tokens they exhibit for each S-invariant.

Let $\mathbf{Iv} = [Iv_1, Iv_2, \ldots, Iv_P]$ be a row vector of P components, with $\mathcal{E}^{[\mathbf{m} \geq \mathbf{Iv}]}(\mathbf{K}, \mathcal{P})$ we denote the set

$$\mathcal{E}^{[\mathbf{m} \geq \mathbf{Iv}]}(\mathbf{K}, \mathcal{P}) = \{\mathbf{m} \in \mathcal{E}(\mathbf{K}, \mathcal{P}) : \mathbf{m} \geq \mathbf{Iv}\}, \tag{11}$$

where $\mathbf{m} \geq \mathbf{Iv}$ is a component-wise inequality.

A second partition of the reachability set is that of grouping together all the markings that are characterized by a constant number of tokens in a given place. In this case with $\mathcal{E}^{[m(p)=l]}(\mathbf{K}, \mathcal{P})$, where $1 \leq p \leq P$ and $l > 0$, we denote the set

$$\mathcal{E}^{[m(p)=l]}(\mathbf{K}, \mathcal{P}) = \{\mathbf{m} \in \mathcal{E}(\mathbf{K}, \mathcal{P}) : m(p) = l\}. \tag{12}$$

From Definition 9 follows the next lemma.

LEMMA 1 [**State Space Lemma**] *For any S-invariant reachable SPN* $(\mathcal{P}, \mathcal{T}, I, O, \mathcal{R}, \mathbf{m_0})$ *with load vector* $\mathbf{K} = \mathbf{S}.\mathbf{m_0^T}$, *the following relations hold:*

$$\mathcal{E}(\mathbf{K},\mathcal{P}) = \bigcup_{j=0}^{Mx_p(\mathbf{K})} \mathcal{E}^{[m(p)=j]}(\mathbf{K},\mathcal{P}); \tag{13}$$

$$\mathcal{E}\left(\mathbf{K}-\mathbf{S.Iv^T},\mathcal{P}\right) = \\ = \left\{\mathbf{m}-\mathbf{Iv} : \mathbf{m} \in \mathcal{E}^{[\mathbf{m}\geq\mathbf{Iv}]}(\mathbf{K},\mathcal{P})\right\}; \tag{14}$$

$$\mathcal{E}\left(\mathbf{K}-l\cdot\mathbf{S_p},\mathcal{P}-\{p\}\right) = \\ = \left\{\mathbf{m}-l\cdot\mathbf{e_p} : \mathbf{m} \in \mathcal{E}^{[m(p)=l]}(\mathbf{K},\mathcal{P})\right\}; \tag{15}$$

where

$$Mx_p(\mathbf{K}) = \left\lfloor \frac{K[r]}{S[r,p]} \right\rfloor$$

is an upper bound for the number of tokens in place p with a load vector $\mathbf{K}$, $K[r]$ is the minimum among the components of the $\mathbf{K}$ vector corresponding to S-invariants that cover place p (with $1 \leq r \leq Q$ where Q is the number of minimal S-invariants), $S[r,p]$ is the entry of the matrix $\mathbf{S}$ corresponding to the r-th S-invariant and to place p, $\mathbf{S_p}$ is the column of the matrix $\mathbf{S}$ corresponding to place p, $\mathbf{Iv}$ is a row vector of P components and l is an integer such that $0 \leq l \leq Mx_p(\mathbf{K})$. $\diamondsuit$

The notation $\mathcal{E}(\mathbf{K},\mathcal{P})$ can be generalized to any subset of places $\wp \subseteq \mathcal{P}$ and any load vector $\mathbf{k} \leq \mathbf{K}$ as follows:

$$\begin{aligned}\mathcal{E}(\mathbf{k},\wp) &= \mathcal{E}(\mathbf{k},\mathcal{P}-\overline{\wp}) \\ &= \{\mathbf{m} \in \mathcal{E}(\mathbf{k},\mathcal{P}) : m(p) = 0 \quad \forall\ p \in \overline{\wp}\},\end{aligned}$$

where $\overline{\wp} = \mathcal{P} - \wp$.

DEFINITION 10 [**Marking Set**] *Given a S-invariant reachable SPN with a load vector $\mathbf{k}$, for any $\wp \subseteq \mathcal{P}$, and $\forall\ p \in \wp$, the set*

$$M_p(\mathbf{k},\wp) = \{i : \exists\ \mathbf{m} \in \mathcal{E}(\mathbf{k},\wp) \text{ and } m(p) = i\} \tag{16}$$

is called the marking set *of p.*

4.2 The Recursive Expressions

In a SPN satisfying the hypothesis of Theorems 1 and 2 we can express the equilibrium distribution as,

$$\pi(\mathbf{m}) = \frac{1}{G}\cdot\prod_{i\in\mathcal{P}} f_i(d)^{m(i)}, \tag{17}$$

and hence,

$$G = \sum_{\mathbf{m} \in \mathcal{E}(\mathbf{K}, \mathcal{P})} \prod_{i \in \mathcal{P}} f_i(d)^{m(i)}.$$

The normalization constant depends on the number of places, and on the load vector $\mathbf{K}$, through the initial marking $\mathbf{m_0}$ (recall that $\mathbf{K} = \mathbf{S}.\mathbf{m_0}$). We can thus express this dependency explicitly by defining an auxiliary function $g(\mathbf{k}, \wp)$, where $\mathbf{k} \leq \mathbf{K}$ and $\wp \subseteq \mathcal{P}$, as

$$g(\mathbf{k}, \wp) = \sum_{\mathbf{m} \in \mathcal{E}(\mathbf{k}, \wp)} \prod_{i \in \wp} f_i(d)^{m(i)}, \tag{18}$$

from which follows that

$$G = g(\mathbf{K}, \mathcal{P}). \tag{19}$$

The following theorem (adapted from [6]) derives a convolution equation by sub–dividing the state space and by conditioning on the number of tokens in one particular place.

THEOREM 3 [**Normalization Constant (I)**] *The following relationship exists between the normalization constant of a PFS S-invariant reachable SPN with a set of places $\mathcal{P}$ and a load vector $\mathbf{K}$, and the normalization constants of SPN with smaller set of places and smaller load vectors. Let be $\mathbf{k}$ and $\wp$ respectively a load vector and a subset of places, with $\mathbf{k} \leq \mathbf{K}$ and $\wp \subseteq \mathcal{P}$, we have that*

$$g(\mathbf{k}, \wp) = \sum_{j \in M_p(\mathbf{k}, \wp)} f_p(d)^j g(\mathbf{k} - j \cdot \mathbf{S_p}, \wp - \{p\}) \qquad p \in \wp \tag{20}$$

where

$$g(\mathbf{0}, \wp) = 1 \tag{21}$$

$$g(\mathbf{k}, \{p\}) = \sum_{j \in M_p(\mathbf{k}, \{p\})} f_p(d)^j. \tag{22}$$

$$g(\mathbf{k}, \wp) = 0 \quad \forall \ \mathbf{k} : \mathcal{E}(\mathbf{k}, \wp) = \emptyset. \tag{23}$$

Proof: Consider the partitioning of the reachability set $\mathcal{E}(\mathbf{k}, \wp)$ given by

$$\mathcal{E}(\mathbf{k}, \wp) = \bigcup_{j \in M_p(\mathbf{k}, \wp)} \mathcal{E}^{[m(p)=j]}(\mathbf{k}, \wp).$$

The auxiliary function can be rewritten in the following manner

$$g(\mathbf{k}, \wp) = \sum_{j \in M_p(\mathbf{k}, \wp)} \left\{ \sum_{\mathbf{m} \in \mathcal{E}^{[m(p)=j]}(\mathbf{k}, \wp)} \prod_{i \in \wp} f_i(d)^{m(i)} \right\},$$

by Condition 15 of Lemma 1 we have that

$$g(\mathbf{k},\wp) = \sum_{j\in M_p(\mathbf{k},\wp)} f_p(d)^j \left\{ \sum_{\mathbf{m}\in\mathcal{E}(\mathbf{k}-j\cdot\mathbf{S_p},\wp-\{p\})} \prod_{i\in\wp-\{p\}} f_i(d)^{m(i)} \right\}$$

$$= \sum_{j\in M_p(\mathbf{k},\wp)} f_p(d)^j g(\mathbf{k}-j\cdot\mathbf{S_p},\wp-\{p\}).$$

When $\mathbf{k}=\mathbf{0}$ the only marking in $\mathcal{E}(\mathbf{k},\wp)$ is the zero marking. Substituting the zero marking in

$$g(\mathbf{k},\wp) = \sum_{\mathbf{m}\in\mathcal{E}(\mathbf{k},\wp)} \prod_{i\in\wp} f_i(d)^{m(i)}$$

we obtain

$$g(\mathbf{0},\wp) = 1.$$

When only place p is left in the sub net it is immediate to derive that

$$g(\mathbf{k},\{p\}) = \sum_{j\in M_p(\mathbf{k},\{p\})} f_p(d)^j.$$

The last boundary condition follows when the load vector $\mathbf{k}$ is such that there is no marking $\mathbf{m}$ in the set $\mathcal{E}(\mathbf{k},\wp)$. $\diamondsuit$

Remark: It must be clear that Theorem 3 is the same as Theorem 4.1 of reference [6]. The reason for including it into this paper is that the introduction of an auxiliary function and of a state partitioning makes, in our opinion, the proof simpler and provides a direct basis for the derivation of our computationally efficient algorithm.

Equation (20) has the structure of a convolution and this is the reason for calling the algorithms derived from equations of this type *Convolution Algorithms.* It is important to note that in the previous theorem we have to compute the marking sets $M_p(\mathbf{k},\wp)$ for each place $p\in\wp$. Finding these marking sets involves the solution of an equal number of systems of linear Diophantine equations. In the following we will show how it is possible to use the special feature of this kind of SPNs (S-invariant reachable) to make the solution of this problem computationally feasible (and simple).

THEOREM 4 [**Normalization Constant (II)**] *Given a S-invariant reachable SPN with PFS and load vector* $\mathbf{K}$. *Then the following relationship exists between the normalization constant of a PFS S-invariant reachable SPN with a set of places* $\mathcal{P}$ *and a load vector* $\mathbf{K}$, *and the normalization constants of SPN with smaller set of places and smaller load vectors. Let be* $\mathbf{k}$ *and* $\wp$ *respectively a load vector and a subset of places, with* $\mathbf{k}\leq\mathbf{K}$ *and* $\wp\subseteq\mathcal{P}$, *we have that*

$$g(\mathbf{k},\wp) = g(\mathbf{k},\wp-\{p\}) + f_p(d)\cdot g(\mathbf{k}-\mathbf{S_p},\wp). \tag{24}$$

Proof: From Theorem 3 we have that

$$g(\mathbf{k},\wp) = \sum_{j\in M_p(\mathbf{k},\wp)} f_p(d)^j g(\mathbf{k}-j\mathbf{S_p},\wp-\{p\}).$$

Let $Mx_p(\mathbf{k})$ be an upper bound for the number of tokens in place p with a load vector $\mathbf{k}$, we call the set

$$CM_p(\mathbf{k},\wp) = \{i : 0 \le i \le Mx_p(\mathbf{k})\}$$

candidate marking set of place p, given $\wp$ and $\mathbf{k}$. Obviously we have that $M_p(\mathbf{k},\wp) \subseteq CM_p(\mathbf{k},\wp)$. For any $i \in \{CM_p(\mathbf{k},\wp) - M_p(\mathbf{k},\wp)\}$ we have that $\mathcal{E}(\mathbf{k}-i\mathbf{S_p},\wp) = \emptyset$, and hence from (23) it follows that

$$\begin{aligned} g(\mathbf{k},\wp) &= \sum_{j=0}^{Mx_p(\mathbf{k})} f_p(d)^j g(\mathbf{k}-j\mathbf{S_p},\wp-\{p\}) \\ &= g(\mathbf{k},\wp-\{p\}) + \sum_{j=1}^{Mx_p(\mathbf{k})} f_p(d)^j g(\mathbf{k}-j\mathbf{S_p},\wp-\{p\}). \end{aligned}$$

Since we have that

$$\begin{aligned} &\sum_{j=1}^{Mx_p(\mathbf{k})} f_p(d)^j g\left(\mathbf{k}-j\mathbf{S_p},\wp-\{p\}\right) = \\ &= f_p(d)\sum_{j=1}^{Mx_p(\mathbf{k})} f_p(d)^{j-1} g\left(\mathbf{k}-j\mathbf{S_p}-(j-1)\mathbf{S_p},\wp-\{p\}\right) \\ &= f_p(d)\sum_{i=0}^{Mx_p(\mathbf{k})-1} f_p(d)^i g(\mathbf{k}'-i\mathbf{S_p},\wp-\{p\}) \\ &= f_p(d)\, g\left(\mathbf{k}',\wp\right), \end{aligned}$$

with $\mathbf{k}' = \mathbf{k}-\mathbf{S_p}$, the theorem is proved. ◇

This recursive expression for $g(\mathbf{k},\wp)$, that we believe to be new, provides the key for the construction of an algorithm to compute the desired normalization constant without having to generate the whole reachability set of the SPN.

4.3 The Algorithm

Similarly to what has been done for the convolution method for Product Form Queueing Networks [4], the operations of our new algorithm can be described with the help of a two dimensional "tableau" with as many columns as there are places in the SPN and as many rows as there are possible load vectors from an empty net to a net with initial marking $\mathbf{m_0}$. The first row and the first column of the tableau are initialised according to Eq. (21) and (23). The tableau

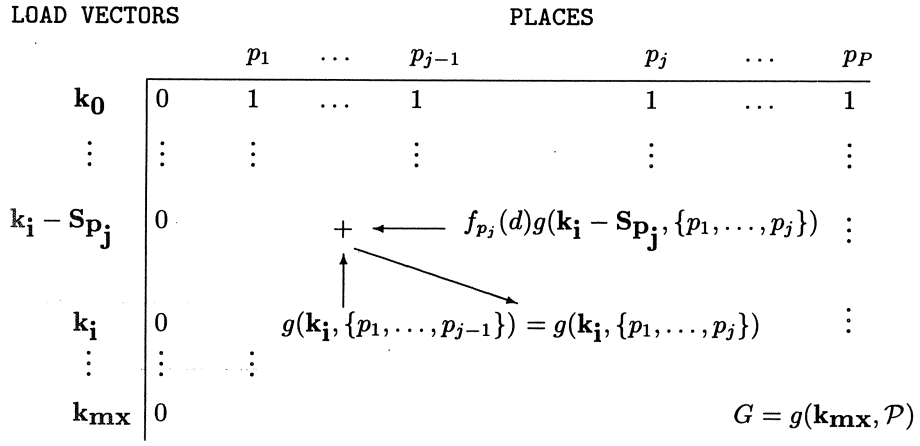


FIGURE 1. Two dimensional "tableau" for computing the normalization constant of a PFS S-invariant reachable SPN.

is filled column-wise starting from the upper left corner. Figure 1 depicts the operations of the convolution algorithm. Let $\mathbf{k_0}, \mathbf{k_1}, \ldots, \mathbf{k_{mx}}$ be a load vector sequence such that $\mathbf{k_0} = [0, 0, \ldots, 0]$, $\mathbf{k_{mx}} = \mathbf{K} = [K_1, K_2, \ldots, K_Q]$ (Q is the number of S-invariants). This is the sequence of all load vectors $\mathbf{k'} \leq \mathbf{K}$. The length of this sequence is $\prod_{i=1}^{Q}(K_i + 1)$. Using Theorems 3 and 4 we can derive the algorithm. At the step corresponding to the set of places $\{p_1, \ldots, p_j\}$, the normalization constant $g(\mathbf{k_i}, \{p_1, \ldots, p_j\})$ can be computed using Theorem 4, and hence $g(\mathbf{k_i}, \{p_1, \ldots, p_j\}) = g(\mathbf{k_i}, \{p_1, p_2, \ldots, p_{j-1}\}) + f_{p_j}(d) \cdot g(\mathbf{k_i - S_{p_j}}, \{p_1, p_2, \ldots, p_j\})$.

The time complexity to compute the normalization constant of a PFS SPN with a load vector $\mathbf{K} = [K_1, K_2, \ldots, K_Q]$ is $O(P \cdot mx)$, where $mx = \prod_{i=1}^{Q}(K_i + 1)$ (Q is the number of S-invariants) and $P = |\mathcal{P}|$. In terms of space the complexity is $O(mx)$.

4.4 Performance Indices using the Normalization Constant

The previous algorithm can be used to find the normalization constant of any S-invariant reachable SPN with PFS. The importance of this method is that the computation of the normalization constant allows an easy evaluation of the SPN since many performance measures can be expressed in terms of normalization constants with smaller set of places and load vectors. In this section we describe how to derive relationships for transition throughputs, place utilizations and average number of tokens in places.

LEMMA 2 **[Transition Utilization]** *Given a S-invariant reachable SPN with PFS, the steady state probability that a transition t is enabled, given a load vector $\mathbf{K}$, is provided by*

$$P([t>; \mathbf{K}) = \frac{g(\mathbf{K} - \mathbf{S.I}(t), \mathcal{P})}{g(\mathbf{K}, \mathcal{P})} \prod_{i \in \mathcal{P}} f_i(d)^{I_i(t)}. \tag{25}$$

ALGORITHM FOR COMPUTING THE NORMALIZATION CONSTANT

```
/* Input:  S, f_(.)(d), k_0, k_1, ..., k_mx
   Output: G */
(1) begin
    /* Declarations */
(2)     g, old_g : array [0, ..., mx] of real

    /* Initialization */
(3)     old_g[0] := 1
(4)     for i := 1 to mx do
(5)         old_g[i] := 0

(6)     for each p ∈ P do
(7)         begin
(8)             g[0] := 1
(9)             for j := 0 to mx do
(10)                begin
(11)                    ind := "index corresponding, in the sequence
                        k_0, k_1, ..., k_mx, to vector load k_j - S_p"
(12)                    g[j] := old_g[j] + f_p(d) · g[ind]
(13)                end
(14)            for j := 0 to mx do
(15)                old_g[j] := g[j]
(16)        end

(17)        G := g[mx]
(18) end
```

FIGURE 2. Algorithm for computing the normalization constant of a PFS S-invariant reachable SPN.

Proof: The steady state probability that a transition t is enabled is given by

$$\begin{aligned} P([t>;\mathbf{K}) &= \sum_{\mathbf{m}\in\mathcal{E}[\mathbf{m}\geq\mathbf{I}(t)](\mathbf{K},\mathcal{P})} \pi(\mathbf{m}) \\ &= \frac{1}{g(\mathbf{K},\mathcal{P})} \sum_{\mathbf{m}\in\mathcal{E}[\mathbf{m}\geq\mathbf{I}(t)](\mathbf{K},\mathcal{P})} \prod_{i\in\mathcal{P}} f_i(d)^{m_i}, \end{aligned}$$

from Condition 14 of Lemma 1, it follows that

$$P([t>;\mathbf{K}) = \frac{1}{g(\mathbf{K},\mathcal{P})} \prod_{i:I_i(t)>0} f_i(d)^{I_i(t)} \sum_{\mathbf{m}\in\mathcal{E}(\mathbf{K}-\mathbf{S}.\mathbf{I}(t),\mathcal{P})} \prod_{i\in\mathcal{P}} f_i(d)^{m_i}$$

$$= \frac{1}{g(\mathbf{K},\mathcal{P})} \prod_{i\in\mathcal{P}} f_i(d)^{I_i(t)} \cdot g(\mathbf{K}-\mathbf{S.I}(t),\mathcal{P})$$
$$= \frac{g(\mathbf{K}-\mathbf{S.I}(t),\mathcal{P})}{g(\mathbf{K},\mathcal{P})} \prod_{i\in\mathcal{P}} f_i(d)^{I_i(t)}$$

where $I_i(t)$ is the i^{th} element of the input bag for transition t. $\Diamond$

Since all transitions have independent marking firing rates, the utilisation and the throughput of a transition t, given a load vector $\mathbf{K}$, are respectively

$$U_t(\mathbf{K}) = P([t>;\mathbf{K}), \tag{26}$$

and

$$X_t(\mathbf{K}) = U_t(\mathbf{K}) \cdot \mu(t). \tag{27}$$

LEMMA 3 **[Place Utilization]** *Given a S-invariant reachable SPN with PFS, the steady state probability that a place p is not empty, given a load vector $\mathbf{K}$, is provided by*

$$P(m(p) > 0;\mathbf{K}) = 1 - \frac{g(\mathbf{K},\mathcal{P}-\{p\})}{g(\mathbf{K},\mathcal{P})}. \tag{28}$$

Proof: The steady state probability that a place p is empty is given by

$$P(m(p) = 0;\mathbf{K}) = \sum_{\mathbf{m}\in\mathcal{E}[m(p)=0](\mathbf{K},\mathcal{P})} \pi(\mathbf{m}),$$

from Condition 15 of Lemma 1 we have that

$$P(m(p) = 0;\mathbf{K}) = \frac{1}{g(\mathbf{K},\mathcal{P})} \sum_{\mathbf{m}\in\mathcal{E}(\mathbf{K},\mathcal{P}-\{p\})} \prod_{i\in\mathcal{P}} f_i(d)^{m_i}$$
$$= \frac{g(\mathbf{K},\mathcal{P}-\{p\})}{g(\mathbf{K},\mathcal{P})}.$$

$\Diamond$

Remark: Lemma 2 and Lemma 3 were proven in [6].

LEMMA 4 **[Place Marking]** *Given a S-invariant reachable SPN with PFS, the steady state probability that in place p there are l tokens, given a load vector $\mathbf{K}$, is provided by*

$$P(m(p) = l;\mathbf{K}) = \frac{g(\mathbf{K}-l\cdot\mathbf{S_p},\mathcal{P}-\{p\})}{g(\mathbf{K},\mathcal{P})} \cdot f_p(d)^l. \tag{29}$$

Proof: The steady state probability that there are l tokens in place p is given by

$$P(m(p) = l;\mathbf{K}) = \sum_{\mathbf{m}\in\mathcal{E}[m(p)=l](\mathbf{K},\mathcal{P})} \pi(\mathbf{m}),$$

from Condition 15 of Lemma 1 we have that

$$P(m(p) = l; \mathbf{K}) = \frac{1}{g(\mathbf{K}, \mathcal{P})} f_p(d)^l \sum_{\mathbf{m} \in \mathcal{E}(\mathbf{K} - l \cdot \mathbf{S_p}, \mathcal{P} - \{p\})} \prod_{i \in \mathcal{P}} f_i(d)^{m_i}$$

$$= \frac{g(\mathbf{K} - l \cdot \mathbf{S_p}, \mathcal{P} - \{p\})}{g(\mathbf{K}, \mathcal{P})} f_p(d)^l.$$

◇

Let us note that $g(\mathbf{K} - l \cdot \mathbf{S_p}, \mathcal{P} - \{p\})$ is the normalization constant of the net with the first $P - 1$ places and a load vector $\mathbf{K} - l \cdot \mathbf{S_p}$.

Theorem 4 gives us an efficient way to compute the distribution of the number of tokens in a place only for the place p_P (the last one of the SPN). Moreover if we have to compute the distribution of the number of tokens in a place $p_i \neq p_P$, we can change the order in which places are considered devising the computation of G putting the place p_i in the last position (place re-indexing). However, this problem can be solved in a computationally more efficient way by using the following inverse convolution algorithm. Let us denote the normalization constant $g(\mathbf{k} - l \cdot \mathbf{S_p}, \wp - \{p\})$ by $g^{[p]}(\mathbf{k} - l \cdot \mathbf{S_p}, \wp)$, where $\wp \subseteq \mathcal{P}$ and $\mathbf{k} \leq \mathbf{K}$.

LEMMA 5 [**Inverse Convolution**] *The following relationship exists between the normalization constant of a PFS S-invariant reachable SPN with a set of places $\mathcal{P}$ and a load vector $\mathbf{K}$, and the normalization constants of SPN with smaller set of places and smaller load vectors. Let be $\mathbf{k}$ and $\wp$ respectively a load vector and a subset of places, with $\mathbf{k} \leq \mathbf{K}$ and $\wp \subseteq \mathcal{P}$, we have that*

$$g^{[p]}(\mathbf{k}, \wp) = g(\mathbf{k}, \wp) - f_p(d) \cdot g(\mathbf{k} - \mathbf{S_p}, \wp). \tag{30}$$

`Proof:` The proof of this lemma follows from Theorem 4. The inverse convolution values can be iteratively computed starting with the initial condition

$$g^{[p]}(\mathbf{0}, \wp) = g(\mathbf{0}, \wp) = 1.$$

◇

It is important to note that the algorithm for the inverse convolution can be numerically unstable. The problem follows from the cancellation error (see [7] for details) that may derive from the subtraction in (30) and suggests that the use of this expression must be carefully controlled in order to avoid unsafe situations in which error becomes predominant [3].

Using Lemma 5 we can compute the average number of tokens in a place p:

$$np_p(\mathbf{K}) = \sum_{i=0}^{K_r} i \cdot P(m(p) = i; \mathbf{K}) \tag{31}$$

$$= \frac{1}{g(\mathbf{K}, \mathcal{P})} \sum_{i=0}^{K_r} i \cdot g^{[p]}(\mathbf{K} - i \cdot \mathbf{S_p}, \mathcal{P}) \cdot f_p(d)^i, \tag{32}$$

where K_r is the maximum among the components of the load vector $\mathbf{K}$ corresponding to S-invariants that cover place p.

Events		Conditions	
Transition	Activity	Place	Condition
t_1	First internal activity	p_1	Server waiting to restart the cyclic sequence
t_2	Second internal activity	p_2	Server waiting to perform second internal activity
t_3	Third internal activity	p_3	Server waiting to perform third internal activity
t_4	Server acquisition by a first class customer	p_4	First class customer requesting a service
t_5	Service of a first class customer	p_5	First class customer waiting to be served
t_6	Server acquisition by a second class customer	p_6	Second class customer requesting a service
t_7	Service of a second class customer	p_7	Second class customer waiting to be served

TABLE 1. Description of the SPN of Figure 3.

5 NUMERICAL RESULTS

The following example shows how to use all previous concepts to find the equilibrium distribution and some performance indices for a SPN satisfying the conditions of Definition 7.

Example

Consider a system composed of n servers, q first class customers and r second class customers. Each server performs two kinds of activities: A sequence of internal activities is started when the server is idle to detect possible faults and to repair them if needed; the external activities correspond instead to servicing customer requests. External operations are divided into an acquisition phase and a service phase. Customers belonging to the first class can be served only if there is an available server waiting to restart its cyclic diagnose/repair sequence of internal operations. Second class customer requests can be served only if there is a server acquired by a first class customer whose service has not been started yet.

Figure 3 shows a SPN that models this system. The servers are represented by the tokens in place p_1, the tokens in places p_4 and p_6 represent respectively the first and the second class customers. In this SPN is represented the system with three internal activities, three servers, two first class customers and the second class customer. Table 1 summarizes the correspondence among transitions, places and states of the system.

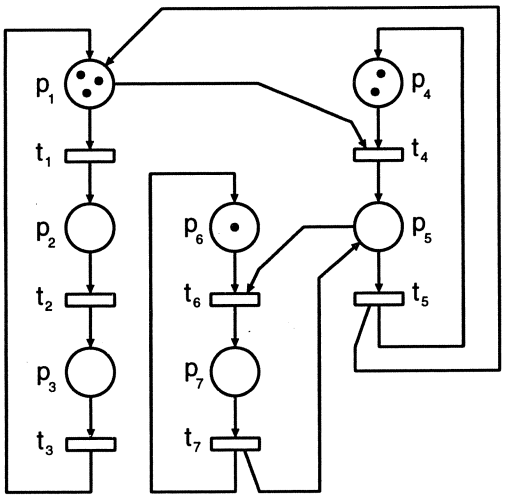


Figure 3. Example SPN used as a test case for the algorithm developed in Section 4.

The incidence matrix $\mathbf{A}$ and the vector $\mathbf{C}(d)$ are given by,

$$\mathbf{A} = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & -1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & -1 \end{pmatrix}$$

and,

$$\mathbf{C}(d) = \begin{bmatrix} log\left(\frac{d(1)}{d(2)}\right) \\ log\left(\frac{d(2)}{d(3)}\right) \\ log\left(\frac{d(3)}{d(1)}\right) \\ log\left(\frac{d(4)}{d(5)}\right) \\ log\left(\frac{d(5)}{d(4)}\right) \\ log\left(\frac{d(6)}{d(7)}\right) \\ log\left(\frac{d(7)}{d(6)}\right) \end{bmatrix}.$$

Let c_i be the i^{th} element of the column vector $\mathbf{C}(d)$. The set $\mathcal{D}$, given by (1), contains functions $d(.)$ such that

$$\begin{aligned}\mu(1)d(1) &= \mu(2)d(2) = \mu(3)d(3) = \alpha_1\\ \mu(4)d(4) &= \mu(5)d(5) = \alpha_2\\ \mu(6)d(6) &= \mu(7)d(7) = \alpha_3\end{aligned}$$

where α_1, α_2, and α_3 are constants; in this case we assume $\alpha_1 = \alpha_2 = \alpha_3 = 1$ and hence $d(i) = \frac{1}{\mu(t_i)}$, for $i = 1, \ldots, 7$.

The SPN has three S-invariants: $[1,1,1,0,1,0,1]$, $[0,0,0,1,1,0,1]$ and $[0,0,0,0,0,1,1]$. The matrix $\mathbf{S}$ is

$$\mathbf{S} = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 1\\ 0 & 0 & 0 & 1 & 1 & 0 & 1\\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

The augmented matrix $[\mathbf{A} \mid \mathbf{C}(d)]$ is row equivalent to the fully row reduced matrix

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_1+c_2+c_3\\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & c_2\\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & c_3\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_4+c_5\\ 1 & 0 & 0 & 1 & -1 & 0 & 0 & c_5\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_6+c_7\\ 0 & 0 & 0 & 0 & 1 & 1 & -1 & c_7 \end{pmatrix}.$$

The three rank conditions are

$$\begin{aligned} c_1+c_2+c_3 &= 0\\ c_4+c_5 &= 0\\ c_6+c_7 &= 0, \end{aligned}$$

which after substituting for the c's are translated in conditions on the $\mu(.)$'s. In this case they are identities

$$\begin{aligned} \frac{\mu(t_2)}{\mu(t_1)}\frac{\mu(t_3)}{\mu(t_2)}\frac{\mu(t_1)}{\mu(t_3)} &= 1\\ \frac{\mu(t_5)}{\mu(t_4)}\frac{\mu(t_4)}{\mu(t_5)} &= 1\\ \frac{\mu(t_7)}{\mu(t_6)}\frac{\mu(t_6)}{\mu(t_7)} &= 1, \end{aligned}$$

and imply that the PFS holds for any possible set of $\mu(.)$'s.

Letting $f_1(d) = f_5(d) = f_7(d) = 1$ gives,

$$h(\mathbf{m}) = \left[\frac{\mu(t_1)}{\mu(t_2)}\right]^{m(2)} \left[\frac{\mu(t_1)}{\mu(t_3)}\right]^{m(3)} \left[\frac{\mu(t_5)}{\mu(t_4)}\right]^{m(4)} \left[\frac{\mu(7)}{\mu(6)}\right]^{m(6)}.$$

Let us show now how it is possible to compute some performance indices for the SPN of Fig. 3. Assume that the following firing rates are associated with the transitions of the net $\mathcal{R} = \{1, 2, 3, 5, 6, 7, 8\}$.

Let $\mathbf{m_0} = [3, 0, 0, 2, 0, 1, 0]$ be the initial marking of the SPN; the load vector in this case is $\mathbf{K} = [3, 2, 1]$. Table 2 corresponds to the "tableau" filled by the algorithm of Figure 2.

Load Vector	p_1	p_2	p_3	p_4	p_5	p_6	p_7
[0,0,0]	1.000	1.000	1.000	1.000	1.000	1.000	1.000
[0,0,1]	0.000	0.000	0.000	0.000	0.000	1.143	1.143
[0,1,0]	0.000	0.000	0.000	1.200	1.200	1.200	1.200
[0,1,1]	0.000	0.000	0.000	0.000	0.000	1.371	1.371
[0,2,0]	0.000	0.000	0.000	1.440	1.440	1.440	1.440
[0,2,1]	0.000	0.000	0.000	0.000	0.000	1.646	1.646
[1,0,0]	1.000	1.500	1.833	1.833	1.833	1.833	1.833
[1,0,1]	0.000	0.000	0.000	0.000	0.000	2.095	2.095
[1,1,0]	0.000	0.000	0.000	2.200	3.200	3.200	3.200
[1,1,1]	0.000	0.000	0.000	0.000	0.000	3.657	4.657
[1,2,0]	0.000	0.000	0.000	2.640	3.840	3.840	3.840
[1,2,1]	0.000	0.000	0.000	0.000	0.000	4.389	5.589
[2,0,0]	1.000	1.750	2.361	2.361	2.361	2.361	2.361
[2,0,1]	0.000	0.000	0.000	0.000	0.000	2.698	2.698
[2,1,0]	0.000	0.000	0.000	2.833	4.667	4.667	4.667
[2,1,1]	0.000	0.000	0.000	0.000	0.000	5.333	7.167
[2,2,0]	0.000	0.000	0.000	3.400	6.600	6.600	6.600
[2,2,1]	0.000	0.000	0.000	0.000	0.000	7.543	10.743
[3,0,0]	1.000	1.875	2.662	2.662	2.662	2.662	2.662
[3,0,1]	0.000	0.000	0.000	0.000	0.000	3.042	3.042
[3,1,0]	0.000	0.000	0.000	3.194	5.556	5.556	5.556
[3,1,1]	0.000	0.000	0.000	0.000	0.000	6.349	8.710
[3,2,0]	0.000	0.000	0.000	3.833	8.500	8.500	8.500
[3,2,1]	0.000	0.000	0.000	0.000	0.000	9.714	14.381

TABLE 2. "Tableau" for the SPN of Fig. 3 up to the load vector $\mathbf{K} = [3, 2, 1]$.

The normalization constant in this particular case is $G = 14.381$. The sparseness of the "tableau" and the presence of many entries with identical values illustrate the behaviour of the algorithm and the peculiarities that distinguish SPNs of the type considered in this paper, from multiclass product form queueing networks.

In particular place p_1, p_2 and p_3 are covered by only one S-invariant and the entries of the corresponding columns are obtained by summing the term corresponding to the same load vector in the previous column with the term corresponding to a load vector with one token less in the first S-invariant. The large number of zero entries in the corresponding columns derives from

the fact that many load vectors are "unfeasible" when a net comprising only these first three places is considered. Place p_5 is covered by two S-invariants; some of the entries of column "p_5" are identical to those of column "p_4": this happens when the load vector is such that removing one token both from the first and the second S-invariant yields a load vector that is unfeasible. Place p_7 is covered by all the three S-invariants and the computation of each entry of the corresponding column depends on values that are quite distant in the tableau.

After computing the normalization constant tableau, the throughput of all the transitions in the net and the average number of tokens in each place have been computed using (27) and (31). The results are presented in Table 3. The throughputs of transitions t_1, t_4, and t_6 provide the rates with which the tests and services for first and second class customers are performed in this net. Having computed the average number of tokens in each place, adding up the throughputs of all the transitions that have a given place in their input bag, and using Little's formula [14], it is possible to obtain the average time spent by tokens in that place. Finally, in order to assess the effectiveness of the proposed algorithm, the solution of the net has been computed for several load vectors (different initial markings) and the results have been compared with those obtained using the package *GreatSPN1.5* [5] (when possible). The computation time of *GreatSPN1.5* and of our new algorithm are reported in Table 4 that summarizes the comparisons showing the convenience of using our new approach when the SPNs are large.

$\mathbf{K} = [3,2,1]$		24 States	
Transition	Throughput	Place	Average numbers of tokens
t_1	0.747020	p_1	1.250066
t_2	0.747020	p_2	0.484967
t_3	0.747020	p_3	0.296423
t_4	2.990066	p_4	1.031457
t_5	2.990066	p_5	0.644040
t_6	2.596026	p_6	0.675497
t_7	2.596026	p_7	0.324503

TABLE 3. Performance Measures for $\mathbf{K} = [3,2,1]$.

All calculations were performed on a SunSparc1 ELC workstation with 16 Mbytes of memory.

6 CONCLUSIONS

In this paper we have shown that the steady state probability distribution of SPNs with product form solution can be efficiently computed using algorithms whose space and time complexities are polynomial in the number of places and

Load Vector	Convolution		GreatSPN1.5	
	Columns	Time	States	Time
[3, 2, 1]	24	0.03	24	0.001
[10, 9, 8]	990	2.12	987	2.01
[30, 20, 10]	7161	1.93	35101	87.3
[40, 30, 20]	26691	7.35	122276	354.9
[50, 40, 30]	64821	17.69	– – –	– – –

TABLE 4. Time Comparisons.

in the initial marking of the SPN. Basic to the derivation of such algorithms is a recursive expression of the normalization constant that is a generalization of that derived by J.P. Buzen for multiple class product form queueing networks with load independent service centers. The main algorithm is characterized by two loops, one over the places of the SPN and the other over the feasible loadings of the SPN. The maximum loading of the SPN is computed from the initial marking using the invariant structure of the SPN. The S-invariants provide also an easily computable upper bound on the number of feasible loadings of the SPN.

The peculiar structure of these new algorithms is based on the constraints deriving from the synchronization conditions that are typical of the SPN.

A numerical example is developed to show the quality of the results that can be obtained with this method. The results obtained with the algorithms are validated with those provided by the package *GreatSPN1.5* [5] using a classical Markov chain solution approach. An initial marking is also considered that yields a state space whose size exceeds the capability of *GreatSPN1.5*.

Several extensions of this work are currently under study. One corresponds to the generalization of the criteria used to identify SPNs with PFS in order to recognize Generalized Stochastic Petri Nets [1] that enjoy this same product form property. Another direction of research consists of using the basic recursive results contained in this paper to develop a Mean Value Analysis method for SPNs similar to that existing for product form queueing networks [16, 15, 3].

REFERENCES

1. M. Ajmone Marsan, G. Balbo, and G. Conte. A class of generalized stochastic Petri nets for the performance analysis of multiprocessor systems. *ACM Transactions on Computer Systems*, 2(1), May 1984.
2. F. Baskett, K. M. Chandy, R. R. Muntz, and F. Palacios. Open, closed and mixed networks of queues with different classes of customers. *Journal of the ACM*, 22(2):248–260, April 1975.
3. S. C. Bruell and G. Balbo. *Computational Algorithms for Closed Queueing Networks*. Elsevier North-Holland, New York, 1980.

4. J. P. Buzen. Computational algorithms for closed queueing networks with exponential servers. *Communications of the ACM*, 16(9):527–531, September 1973.
5. Giovanni Chiola. *GreatSPN* 1.5 software architecture. In *Proc.* 5^{th} *Int. Conf. Modeling Techniques and Tools for Computer Performance Evaluation*, Torino, Italy, February 1991.
6. J.L. Coleman, W. Henderson, and P.G. Taylor. Product form equilibrium distributions and an algorithm for classes of batch movement queueing networks and stochastic Petri nets. Technical report, University of Adelaide, 1992.
7. C. E. Froberg. *Introduction to Numerical Analysis.* Addison-Wesley, Reading, MA, 1974.
8. W. J. Gordon and G. F. Newell. Closed queueing systems with exponential servers. *Operations Research*, 15:254–265, 1967.
9. W. Henderson, D. Lucic, and P.G. Taylor. A net level performance analysis of stochastic Petri nets. *Journal of Australian Mathematical Soc. Ser. B*, 31:176–187, 1989.
10. W. Henderson and P.G. Taylor. Aggregation methods in exact performance analysis of stochastic Petri nets. In *Proc. 3rd Intern. Workshop on Petri Nets and Performance Models*, pages 12–18, Kyoto, Japan, December 1989. IEEE-CS Press.
11. W. Henderson and P.G. Taylor. Embedded processes in stochastic Petri nets. *IEEE Transactions on Software Engineering*, 17(2), February 1991.
12. J. R. Jackson. Jobshop-like queueing systems. *Management Science*, 10(1):131–142, October 1963.
13. A.A. Lazar and T.G. Robertazzi. Markovian Petri net protocols with product form solution. *Performance Evaluation*, 12:67–77, 1991.
14. J. D. C. Little. A proof of the queueing formula $L = \lambda W$. *Operations Research*, 9:383–387, 1961.
15. M. Reiser and S. S. Lavenberg. Mean value analysis of closed multichain queueing networks. *Journal of the ACM*, 27(2):313–322, April 1980.
16. K. C. Sevcik and I. Mitrani. The distribution of queueing network states at input and output instants. *Journal of the ACM*, 28(2):358–371, April 1981.
17. M. Silva. *Las Redes de Petri en la Automatica y la Informatica.* Ed. AC, Madrid, Spain, 1985.
18. E. Teruel and M. Silva. Liveness and home states in equal conflict systems. In *Application and Theory of Petri Nets 1992, Proc.* 14^{th} *International Conference*, Chicago, USA, June 1993. LNCS No 616, Springer Verlag.

Operational Analysis of Timed Petri Nets and Application to the Computation of Performance Bounds

G. Chiola*, C. Anglano
Dipartimento di Informatica, Università di Torino
Torino, Italy 10149
J. Campos†, J.M. Colom†, M. Silva†
Dpto. de Ingeniería Eléctrica e Informática
Universidad de Zaragoza
Zaragoza, Spain 50015

We use operational analysis techniques to partially characterize the behaviour of timed Petri nets under very weak assumptions on their timing semantics. New operational inequalities are derived that are typical of the presence of synchronization and that were therefore not considered in queueing network models. We show an interesting application of the operational laws to the statement and the efficient solution of problems related to the estimation of performance bounds insensitive to the timing probability distributions. The results obtained generalize and improve in a clear setting results that were derived in the last few years for several different subclasses of timed Petri nets. In particular the extension to Well-Formed Coloured nets appears straightforward and allows an efficient exploitation of models symmetries.

1 INTRODUCTION

Operational analysis is a conceptually very simple way of deriving mathematical equations relating observable quantities in queueing systems [11]. In [10] the reader can find some nice examples of how the application of operational analysis techniques can help in explaining and proving fundamental results in queueing network analysis. Here we apply operational analysis techniques to derive linear equations and inequalities relating interesting performance measures in timed Petri net models. The main conceptual difference between queueing and Petri net models is the presence of a synchronization primitive in the

*This work was performed while G. Chiola was visiting the University of Zaragoza, Spain, in the framework of the European Grant BRA-QMIPS of CEC DG XIII.
†This work has been supported by the Spanish PRONTIC 354/91 and the Aragonese CONAI-DGA P-IT 6/91.

latter. Early works on extensions of operational analysis to Petri nets include [12], where however synchronization was neglected. New operational inequalities are derived here for synchronization elements that have no counterpart in operational laws for queueing networks.

Some classical results of queueing networks were already proven to hold in stochastic Petri net models. In this paper we derive, under much weaker conditions, a generalization of the classical *utilization law* for the case of multiply enabled transitions and several inequalities that relate throughput, average marking, and average transition firing time in case of synchronization transitions. All these results are derived for each possible observable sample path. Therefore, in order to compare to classical queueing laws stated in a stochastic framework, the additional hypothesis of unique limit behaviour for each sample path must be assumed.

In addition to the mathematical interest of these derivations, we propose also an application of these results to the computation of performance bounds based on linear programming techniques. Such performance bounds are fairly inexpensive to compute compared to the cost of discrete event simulation or exact Markovian analysis, and moreover provide results that are insensitive of the probability distribution of the transition firing times. The linear programming problems (LPP's) presented in this paper represent also a generalization of some recent results published in [3, 4, 2] since they can be applied to arbitrary Petri net structures and reduce to the previous ones when the Petri net structure satisfies some particular constraints.

The paper is organized as follows. Section 2 presents the operational analysis of timed Petri nets and the derivation of the main equations. Section 3 shows the application of the operational laws, also considering the case of Well-Formed Coloured nets, to the statement of LPP's for the computation of performance bounds depending only on the average transition firing times, the structure, and the initial marking of the net. Section 4 provides an example of computation of such bounds in the case of a Coloured Well-Formed timed Petri net model. Finally, Section 5 contains some concluding remarks and ideas for future research on the topics.

2 Observable Quantities and Operational Laws

In this section we start by defining measurable quantities that characterize the state and the behaviour in time of a Petri net model. Then we derive and prove in a very simple and direct way some fundamental relations that hold true "operationally" among them, i.e. that are verified in any sample path that one can measure in an experiment.

We assume the reader to be familiar with the Petri net formalism and notation. We refer to [13] or [15] for an introduction to Petri nets and most of their behavioural properties and analysis techniques. We also refer to [1] for a detailed discussion of different timing semantics and related operation mechanisms. We just resume here the notation conventions that are used in the following of this paper.

$\mathcal{N} = (P, T, W, M_0)$ is a net system, where P is the set of places, T is the set of transitions, $W : P \times T \cup T \times P \rightarrow \mathbb{N}$ is the incidence function, and M_0 is the initial marking (in general, a marking is $M : P \rightarrow \mathbb{N}$, and $\forall p_i \in P$, $M[p_i]$ is the number of tokens in p_i). The input (output) set of $x \in P \cup T$ is ${}^\bullet x = \{y \in P \cup T \mid W(y,x) \geq 1\}$ ($x^\bullet = \{y \in P \cup T \mid W(x,y) \geq 1\}$).

2.1 Basic operational quantities

Assume that a generic timed Petri net is available for measurement, and that the following quantities can be collected during an experiment, starting at time $\tau = 0$ and ending at time $\tau = \theta > 0$, at which all transitions have been fired at least once. The total number of transitions firings during the experiment is assumed finite.

Instantaneous marking: $\forall p_k \in P$, $\forall \tau : 0 \leq \tau \leq \theta$, $M[p_k](\tau)$ represents the number of tokens in place p_k at time τ.

Average marking during the experiment interval:

$$\forall p_k \in P, \quad \bar{M}[p_k](\theta) = \frac{1}{\theta} \int_0^\theta M[p_k](\tau) d\tau$$

Instantaneous enabling degree: $\forall t_i \in T, \forall \tau : 0 \leq \tau \leq \theta$, $e_i(\tau)$ represents the internal concurrency of transition t_i at time τ, i.e.

$$e_i(\tau) = \max\{k \in \mathbb{N} : \forall p \in {}^\bullet t_i,\ M[p](\tau) \geq k\ W(p, t_i)\}$$

The following relation holds by definitions:

$$\forall t_i \in T, \quad \forall \tau, \quad e_i(\tau) = \min_{p \in {}^\bullet t_i} \left\lfloor \frac{M[p](\tau)}{W(p, t_i)} \right\rfloor \tag{1}$$

(where $\forall a \in \mathbb{R}$, $\lfloor a \rfloor$ denotes the largest integer not greater than a).

Average enabling degree: $\forall t_i \in T$, $\bar{e}_i(\theta) = \frac{1}{\theta} \int_0^\theta e_i(\tau) d\tau$ represents the average number of servers active in transition t_i during the experiment interval.

Since we use an "infinite-server" semantics for transition enabling, we need to consider the activities of the different servers in a given transition t_i independently. Without loss of generality we assume an ordering of the servers associated with transitions such that busy servers always come before idle servers, i.e., at any point in time τ the first $e_i(\tau)$ servers are active inside transition t_i, while the remaining ones are idle.

Under this assumption we can define the:

Number of firings completed by the j-th server in t_i from time 0 up to time θ, denoted $F_{i,j}(\theta)$.

Total number of firings of t_i during the experiment interval

$$F_i(\theta) = \sum_{j=1}^{\infty} F_{i,j}(\theta)$$

(by assumptions, $0 < F_i(\theta) < \infty$).

Throughput of t_i $x_i(\theta) = \frac{F_i(\theta)}{\theta}$ that represents the average number of firings completed per time unit.

2.2 Conflict-free nets

In case of nets without conflicts one can easily define the average service time of transitions as a function of the busy times of all servers. In particular we define:

Instantaneous enabling of j-th server in t_i

$$e_{i,j}(\tau) = \text{ if } e_i(\tau) \geq j \text{ then 1 else 0}$$

characteristic function that evaluates to 1 if and only if the j-th server in transition t_i is busy at time τ.

Busy time for the j-th server of t_i $\theta_{i,j}(\theta) = \int_0^{\theta} e_{i,j}(\tau)d\tau$

Service time for the j-th server of t_i $S_{i,j}(\theta) = \frac{\theta_{i,j}(\theta)}{F_{i,j}(\theta)}$

Average service time for t_i $\bar{S}_i(\theta) = \frac{\sum_{j=1}^{\infty} \theta_{i,j}(\theta)}{\sum_{j=1}^{\infty} F_{i,j}(\theta)}$

The following equation holds for any measurement experiment:

Enabling operational law

$$\forall t_i \in T, \quad \bar{e}_i(\theta) = x_i(\theta)\bar{S}_i(\theta) \tag{2}$$

Proof: By definition $x_i(\theta) = \frac{\sum_{j=1}^{\infty} F_{i,j}(\theta)}{\theta}$ then multiplying and dividing by $\sum_{j=1}^{\infty} \theta_{i,j}(\theta)$ and recalling the definition of $\bar{S}_i(\theta)$ we obtain:

$$\forall t_i \in T, \quad x_i(\theta)\bar{S}_i(\theta) = \frac{1}{\theta} \sum_{j=1}^{\infty} \theta_{i,j}(\theta)$$

and then substituting the definition of $\theta_{i,j}(\theta)$ and exchanging the integral and the sum signs:

$$\forall t_i \in T, \quad x_i(\theta)\bar{S}_i(\theta) = \frac{1}{\theta}\int_0^\theta \sum_{j=1}^{\infty} e_{i,j}(\tau)d\tau$$

Now it is trivial to identify the integrand to be the instantaneous enabling degree $e_i(\tau)$, so that the result follows. Q.E.D.

The above enabling law is the well-known "utilization law" derived in the framework of multiple server queues. From the enabling law it follows that if the average firing time of a transition is known, then its throughput is proportional to its average enabling degree. Of course in case of immediate transitions $\bar{S}_i(\theta) = 0$, so immediate transitions are never enabled for non-null intervals of time.

We are now in a position to state our *synchronization inequalities* that relate the throughput and the average marking of the input places for any transition.

Upper bound inequality. $\forall t_i \in T$,

$$x_i(\theta)\ \bar{S}_i(\theta)\ \leq\ \min_{p_k \in {}^\bullet t_i}\left(\frac{\bar{M}[p_k](\theta)}{W(p_k,t_i)}\right) \tag{3}$$

The inequality becomes an equality whenever $\sum_{p\in {}^\bullet t_i} W(p,t_i) = 1$.

Proof: We start from Equation (1) that is valid in each instant of the experiment. Of course this implies that $\forall p_k \in {}^\bullet t_i, \forall \tau\ :\ 0 \leq \tau \leq \theta,\ e_i(\tau) \leq \frac{M[p_k](\tau)}{W(p_k,t_i)}$. Therefore $\forall p_k \in {}^\bullet t_i,\ \bar{e}_i(\theta) \leq \frac{\bar{M}[p_k](\theta)}{W(p_k,t_i)}$, and applying the enabling operational law the result follows. Q.E.D.

This inequality establishes an upper bound for the average enabling (hence for the transition throughput once the service time is defined) in the case of transitions with more than one input place that model a synchronization. In the following we derive other inequalities that establish lower bounds as well. We shall see that in the particular case of transitions with a single input place the two inequalities reduce to a single equality.

Lower bound inequality for single input arc ($W(p,t_i) \geq 1$). $\forall t_i \in T\ :\ {}^\bullet t_i = \{p\}$,

$$x_i(\theta)\ \bar{S}_i(\theta)\ \geq\ \frac{\bar{M}[p](\theta) - W(p,t_i) + 1}{W(p,t_i)} \tag{4}$$

Notice that in case $W(p,t_i) = 1$ this reduces to $x_i(\theta)\ \bar{S}_i(\theta) \geq \bar{M}[p](\theta)$, that combined with the upper bound inequality (3) reduces to the equation $x_i(\theta)\ \bar{S}_i(\theta) = \bar{M}[p](\theta)$.

Proof: First define some auxiliary punctual marking functions:

$$\forall p \in P,\ \forall \tau, \quad M^v[p](\tau)\ =\ \max(0, M[p](\tau) - v)$$

$$\forall p \in P,\ \forall \tau, \quad M_l^u[p](\tau)\ =\ M^l[p](\tau) - M^u[p](\tau)$$

Consider now the following properties of the auxiliary function $\forall k \in \mathbb{N} : k > 0$,

$$0 \leq M_{kw-1}^{(k+1)w-1}[p](\tau) \leq w$$

Moreover, notice that the k-th server in transition t_i is enabled if and only if $M_{kw-1}^{(k+1)w-1}[p](\tau) \geq 1$ in case $w = W(p,t_i)$. Therefore we can conclude that:

$$\forall t_i \in T : {}^\bullet t_i = \{p\}, \quad \forall k \geq 1, \quad \forall \tau,$$
$$e_{i,k}(\tau) \geq \frac{1}{W(p,t_i)} M_{kW(p,t_i)-1}^{(k+1)W(p,t_i)-1}[p](\tau)$$

Hence we derive: $\forall t_i \in T : {}^\bullet t_i = \{p\}$,

$$\forall \tau, \; e_i(\tau) \geq \frac{1}{W(p,t_i)} \sum_{k=1}^{\infty} M_{kW(p,t_i)-1}^{(k+1)W(p,t_i)-1}[p](\tau) =$$
$$= \frac{M[p](\tau) - W(p,t_i) + 1}{W(p,t_i)}$$

Finally, taking the average over the experiment interval and applying the enabling law, the result follows. Q.E.D.

Observe that in the case that the right-hand expression in (4) is negative, a trivial inequality can be used: $x_i(\theta)\bar{S}_i(\theta) \geq 0$.

Improvement for bounded nets: $\forall t_i \in T : {}^\bullet t_i = \{p\}$, if $\forall \tau$, $M[p](\tau) \leq B_p$ and $w_{ip} = W(p,t_i)$ and $\exists k \in \mathbb{N} : w_{ip}k \leq B_p < (k+1)w_{ip}$

$$x_i(\theta) \; \bar{S}_i(\theta) \geq k \, \frac{\bar{M}[p](\theta) - w_{ip}k + 1}{B_p - w_{ip}k + 1} \tag{5}$$

Proof: Firstly note that $\forall t_i \in T$, $\forall j \in \mathbb{N}$

$$\bar{e}_i(\theta) \geq j \; \bar{e}_{i,j}(\theta) = j \, \frac{\theta_{i,j}(\theta)}{\theta}$$

Secondly, note that the marking in the input place p can be expressed as the sum of two components:

$$\forall \tau, \quad M[p](\tau) = w_{ip} \sum_{j=1}^{\infty} e_{i,j}(\tau) + N[p](\tau)$$

where the component $N[p](\tau) \leq w_{ip} - 1$ represents the portion of marking not used to enable the transition. Now taking the integral and dividing by θ one obtains:

$$\bar{M}[p](\theta) = w_{ip} \; \bar{e}_i(\theta) + \bar{N}[p](\theta)$$

This equation shows that the average enabling depends only on the mean values of the input place marking and of the unused portion of the marking.
The worst case from the point of view of enabling the transition k times occurs when the place is marked with $w_{ip}k - 1$ tokens most of the time and with B_p tokens for the rest of the time, since this case maximizes the unused portion of the average marking in the input place. From these considerations the result follows. Q.E.D.

Lower bound inequality for binary synchronization with ordinary arcs. $\forall t_i \in T : {}^\bullet t_i = \{p_1, p_2\}$ and $W(p_1, t_i) = W(p_2, t_i) = 1$, if $M[p_1](\tau) \leq B_1$ and $M[p_2](\tau) \leq B_2$ and $B_1 \leq B_2$ then

$$x_i(\theta)\ \bar{S}_i(\theta)\ \geq\ \bar{M}[p_1](\theta) + \frac{B_1}{B_2}\bar{M}[p_2](\theta) - B_1 \tag{6}$$

Proof: Similarly to the previous case we can write two equations relating the average marking, the average enabling, and the average portion of unused marking for each of the two input places:

$$\bar{e}_i(\theta)\ =\ \bar{M}[p_1](\theta) - \bar{N}[p_1](\theta)$$

$$\bar{e}_i(\theta)\ =\ \bar{M}[p_2](\theta) - \bar{N}[p_2](\theta)$$

Now we can compute upper bounds on the unused part of the marking as follows. The maximum fraction of time during which $N[p_1](\tau)$ may be greater than zero is equal to the minimum time during which $M[p_2](\tau) = 0$ (otherwise the transition would be enabled and the marking of p_1 would contribute to the enabling instead); since place p_2 is B_2 bounded, this fraction of time is less than or equal to $1 - \frac{\bar{M}[p_2](\theta)}{B_2}$; moreover during this maximum time, the maximum value of the marking in p_1 is less than or equal to B_1. Hence

$$\bar{N}[p_1](\theta)\ \leq\ B_1\ \left(1 - \frac{\bar{M}[p_2](\theta)}{B_2}\right)$$

and from this the result follows trivially. Q.E.D.

A general lower bound for bounded nets: $\forall t_i \in T : {}^\bullet t_i = \{p_1, p_2, \ldots, p_n\}$, $\forall j \leq n$, $M[p_j](\tau) \leq B_j$ and $B_1 \leq B_j$

$$x_i(\theta)\ \bar{S}_i(\theta)\ \geq\ \frac{\bar{M}[p_1](\theta) - W(p_1, t_i) + 1 - B_1\ \max(f_j)}{W(p_1, t_i)} \tag{7}$$

where $\forall j\ :\ 2 \leq j \leq n$, $f_j = 1 - \frac{\bar{M}[p_j](\theta) - W(p_j, t_i) + 1}{B_j - W(p_j, t_i) + 1}$

Proof: Similar to the previous ones writing the upper bound for the quantity $\bar{N}[p_1](\theta)$. Q.E.D.

2.3 General nets with conflicts

In the general case in which transitions may be enabled in conflict the definitions of service time and average enabling degree must be modified in order to take the possibility of preemption into account. In the literature two types of timed Petri net semantics have been proposed: *race* and *preselection* conflict resolution policies [1]. According to the race policy all enabled transitions start working, and the first one that completes its firing time seizes the tokens from the input places, thus possibly preempting other transitions. Instead, the preselection policy requires that conflicts be solved at the enabling time instant,

so that only selected transitions put their servers to work and fire for sure after the elapsing of their firing time. In any case the same kind of results can be derived.

Conditional instantaneous enabling of j-th server in t_i:

$e'_{i,j}(\tau)$ = if "$e_i(\tau) \geq j$ and the enabling is not preempted" then 1 else 0,

characteristic function that evaluates to 1 if and only if the j-th server in transition t_i is busy at time τ and its work will not be wasted due to the preemption from a conflicting transition. Of course $e'_{i,j}(\tau) \leq e_{i,j}(\tau)$ by definition.

Useful busy time for the j-th server of t_i $\theta'_{i,j}(\theta) = \int_0^\theta e'_{i,j}(\tau)d\tau$

Useful service time for the j-th server of t_i $S'_{i,j}(\theta) = \frac{\theta'_{i,j}(\theta)}{F_{i,j}(\theta)}$

Useful average service time for transition t_i $\bar{S'}_i(\theta) = \frac{\sum_{j=1}^{\infty} \theta'_{i,j}(\theta)}{\sum_{j=1}^{\infty} F_{i,j}(\theta)}$

The enabling operational law is extended as:

$$\forall t_i \in T, \quad \bar{e'}_i(\theta) = x_i(\theta)\ \bar{S'}_i(\theta) \tag{8}$$

and the proof is similar to the one shown above. From the comparison with Equation 2 it also follows that $\bar{S'}_i(\theta) \leq \bar{S}_i(\theta)$ independently of the probability distribution of the firing time processes.

Equation (1) however becomes an inequality in case of nets with conflicting transitions:

$$\forall t_i \in T, \ \forall \tau, \quad e'_i(\tau) \leq \min_{p_k \in {}^\bullet t_i} \left(\frac{M[p_k](\tau)}{W(p_k, t_i)} \right) \tag{9}$$

The upper bound inequality (3) still holds in this more general setting by just substituting $S'_{i,j}(\theta)$ for $S_{i,j}(\theta)$.

2.3.1 Race versus preselection policy

The quantities $\bar{e'}_i(\theta)$ and $\bar{S'}_i(\theta)$ are in general measurable from an off-line processing of an experiment record without any further assumption.

Using the preselection policy, the useful service time of a transition is exactly the transition firing time. This allows one to derive an improved version of the upper bound inequality: $\forall p_k \in P$,

$$\sum_{t_i \in p_k^\bullet} \left(W(p_k, t_i)\ x_i(\theta)\ \bar{S}_i(\theta) \right) \ \leq \ \bar{M}[p_k](\theta) \tag{10}$$

In the case of race policy, instead, the useful average service time $\bar{S'}_i(\theta)$ might be strictly less than the nominal transition firing times due to the effect of preemption from conflicting transitions. Inequality (10) holds true in a race policy model only if all transitions that are output for place p_k are *behaviourally persistent* (i.e. their enabling is mutually exclusive). In other words, only the following modified version of inequality (3) holds true for behaviourally conflicting timed transitions with race policy:

$$\forall t_i \in T, \quad x_i(\theta)\ \bar{S'}_i(\theta)\ \leq \min_{p_k \in {}^\bullet t_i} \left(\frac{\bar{M}[p_k](\theta)}{W(p_k, t_i)} \right) \tag{11}$$

For what concerns the synchronization lower bounds, inequalities (4–7) in general apply only to persistent or immediate transitions (in the latter case $\bar{S}_i = \bar{S'}_i = 0$). The case of conflicting transitions with preselection policy may be treated by net transformation as follows, while for the case of conflicting timed transitions with race policy no synchronization lower bound inequality applies.

Consider transition t_i timed, potentially in conflict with other timed transitions and with preselection conflict resolution policy. Split t_i in two transitions t'_i and t''_i and add a new place p'_i such that $\forall p \in P\ W(p, t'_i) = W(p, t_i)$ and $\forall p \in P\ W(t''_i, p,) = W(t_i, p)$ and $W(t'_i, p'_i) = W(p'_i, t''_i) = 1$ and t'_i is immediate and $\bar{S}_{t''_i} = \bar{S}_i$. In the transformed net t''_i is persistent with single input arc (by construction), so that Inequality (4) applies. Transition t'_i is instead immediate, so that a subset of inequalities (4–7) applies even in presence of conflict.

3 Performance Bounds Based on Operational Laws

The inequalities that we derived in the previous section can be used to compute upper and lower bounds for the throughput of transitions or for the average marking of places for general timed Petri nets using linear programming techniques. The idea is to compute vectors $\bar{M}$ and $\vec{x}$ that maximize or minimize the throughput of a transition or the average marking of a place among those verifying the previous operational laws and other linear constraints that can be easily derived from the net structure.

A first set of linear equality constraints can be derived from the fact that the vector $\bar{M}$ is an average weight of reachable markings: $\bar{M} = \sum_{M_r \in RS(\vec{M}_0)} \beta_r M_r$. Since for each reachable marking $M_r = M_0 + C \cdot \vec{\sigma}_r$, we obtain that also the average marking must satisfy the same linear equation:

$$\bar{M} = M_0 + C \cdot \vec{\sigma} \tag{12}$$

where $\vec{\sigma} = \sum_{M_r \in RS(\vec{M}_0)} \beta_r \vec{\sigma}_r$.

The following set of linear inequalities imposes that for each place the token flow out is less than or equal to the token flow in: $\forall p_k \in P$,

$$\sum_{t_i \in {}^\bullet p_k} x_i W(t_i, p_k) \geq \sum_{t_o \in p_k^\bullet} x_o W(p_k, t_o) \tag{13}$$

If place p_k is known to be bounded, then the above inequality becomes an equality which represents the classical *flow balance* equation: $C[p_k] \cdot \vec{x} = 0$.

On the other hand, for each pair of transitions t_i, t_j in (behavioural) free conflict (i.e., such that they are always simultaneously enabled or disabled) the following equation is verified:

$$\frac{x_i}{\alpha_i} = \frac{x_j}{\alpha_j} \tag{14}$$

where α_i, α_j are the routing rates that define the resolution of the conflict between t_i and t_j.

Additionally, most of the operational inequality laws that were derived in the previous section linearly relate the average marking of places with the throughput of their output transitions. Hence they can be considered as constraints for an LPP.

3.1 Extension to TWN's

For timed Well-Formed Coloured nets (TWN's) [8] it is possible to derive, directly from the inequalities developed in the previous sections, operational relations allowing an efficient computation of performance bounds. Given a TWN, the basic idea is to consider the corresponding unfolded net and to apply the relations developed in the previous sections. The relations for the TWN are then obtained combining the partial results for the unfolded one.

A fundamental property that TWN's must have in order to be able to combine the results for the unfolded one is the *symmetry*, meaning that in the unfolded nets obtained from the Well-Formed ones all colour instances of a given place and of a given transition must be equivalent. To be more precise, if a transition t has average service time $\bar{S}_t$, then all of its instances have the same average service time. Moreover if a place p is bounded, then we assume that the maximum number of tokens that each of its instances can contain is the same.

In the rest of this section we show, as an example, the derivation of lower bound inequality for single input arc for TWN's. More details on the derivations can be found in [7].

3.1.1 Notation

In this section we give some notations used in the derivations of relations for TWN's ([8]).

Generic function $f = \sum_{j=1}^{k} F_j$, where F_j is the j^{th} tuple and its arity l is given by the number of colour classes composing the colour domain of the place. This definition of function is slightly different from the classical one, since here we allow linear combinations only outside the tuples (i.e. each tuple is composed only by elementary functions). For example the function $F =< S - x, y >$ is written as $F' =< S, y > - < x, y >$.

Cardinality of function $|f| = \sum_{j=1}^{k} |F_j|$, where $|F_j| = \alpha_j \times \prod_{i=1}^{l} |(F_j)_i|$ is the cardinality of the j^{th} tuple. The coefficient α_j denotes the product of the

coefficients of the elementary functions composing the tuple and $(F_j)_i$ is the i^{th} function of the j^{th} tuple. For example if $F_j =< 3x, 2y >$, then $\alpha_j = 6$.
Family of arcs Each tuple F_j of a function f identifies a set of arcs (with weight α_j), whose cardinality is $A(F_j) = \prod_{i=1}^{l} |(F_j)_h|$. The global number of arcs corresponding to function f is $A(f) = \sum_{j=1}^{k} A(F_j)$, where each $A(F_j)$ has the sign of the corresponding tuple F_j. When $A(f) = 1$, then we denote as α_f the weights associated to the unique family of arcs corresponding to f.
Input and Output relations If t is an input transition of place p (with function f), then $IN(p,t) = \frac{|C(t)|}{|C(p)|} A(f)$ is the number of input instances of t for each instance of p. Similarly if t is an output transition of place p, then $OUT(p,t) = \frac{|C(t)|}{|C(p)|} A(f)$ is the number of output instances of t for each instance of p.

3.1.2 Lower bound inequality for single input arc

To apply this inequality to an unfolded net, the conditions for its applicability must be met for all transition instances. This means that each instance t_i of a coloured transition t must have only one input place. This condition is met if the function f labelling the arc contains only *projection* and *successor* elementary functions (that is $A(f) = 1$).
Inequality for single input arc

$$\forall t \in T : {}^\bullet t = \{p\}, W^-(p,t) = f, \ A(f) = 1$$

$$\alpha_f x_t \bar{S}_t \geq OUT(p,t)\bar{M}[p] - |C(t)|\,(\alpha_f - 1)$$

Proof: Assume to have a portion of a TWN containing transition t and its input place p and that $|C(t)| = n$ and $|C(p)| = m$. Considering the n instances of t we can write the following set of inequalities

$$\forall i \in \{1, \ldots, n\} \ \ \alpha_f x_{t_i} \bar{S}_{t_i} \geq \bar{M}[p_{t_i}] - \alpha_f + 1$$

where p_{t_i} is the unique input place of transition instance t_i. Summing the left-hand sides and the right-hand sides of the above inequalities we obtain:

$$\alpha_f x_t \bar{S}_t \geq (\sum_{i=1}^{n} \bar{M}[p_{t_i}] - |C(t)|\,(\alpha_f - 1)) \tag{15}$$

Since each instance of p appears exactly $OUT(p,t)$ times in the summation of the above expression we can rewrite inequality (15) as

$$\alpha_f x_t \bar{S}_t \geq (OUT(p,t) \sum_{i=1}^{m} M[p_i] - |C(t)|\,(\alpha_f - 1)) \tag{16}$$

and the result follows. Q.E.D.
In a similar way it is possible to derive, for TWN's, the equivalent of relations devised for timed Petri nets.

3.2 *LPP formulation*

Performance bounds for TWN's can be computed solving the LPP of table 1 (whose constraints are the relations derived in the previous sections) where f is a linear function of $\bar{M}$ and $\vec{x}$. The linear programming problem for bounds computation for non coloured timed Petri nets can be obtained from that of table 1 setting $OUT(p,t) = \mid C(p) \mid = \mid C(t) \mid = 1$, $\forall p \in P, t \in T$ and observing that condition $A(f) = 1$ always holds true.

maximize [or minimize] $f(\bar{M}, \vec{s})$	subject to
$\bar{M}[p] = M_0[p] + \sum_{t_i \in {}^\bullet p} \mid f_i \mid \sigma_{t_i} - \sum_{k_j \in p^\bullet} \mid g_j \mid \sigma_{k_j}$;	(c_1) $\forall p \in P : W^+(p,t_i) = f_i,\ W^-(p,k_i) = g_j$
$\sum_{t_i \in {}^\bullet p} \mid f_i \mid x_{t_i} \geq \sum_{k_j \in p^\bullet} \mid g_j \mid x_{k_j}$;	(c_2) $\forall p_k \in P : W^+(p,t_i) = f_i,\ W^-(p,k_i) = g_j$
$\sum_{t_i \in {}^\bullet p} \mid f_i \mid x_{t_i} = \sum_{k_j \in p^\bullet} \mid g_j \mid x_{k_j}$;	(c_2') $\forall p_k \in P$ bounded
$\frac{x_i}{\alpha_i} = \frac{x_j}{\alpha_j}$,	(c_3) $\forall t_i, t_j \in T$: behaviourally free choice
$\mid f \mid \cdot x_t \bar{S}_t \leq OUT(p,t)\bar{M}[p]$	(c_4) $\forall t \in T, \forall p \in {}^\bullet t : W^-(p,t) = f$
$\alpha_f x_t \bar{S}_t \geq OUT(p,t)\bar{M}[p] - \mid C(t) \mid (\alpha_f - 1)$	(c_5) $\forall t \in T$ persistent or immediate : ${}^\bullet t = \{p\}$, $W^-(p,t) = f,\ A(f) = 1$
$x_t \bar{S}_t \geq k \frac{OUT(p,t)\bar{M}[p] + \mid C(t) \mid (1 - k\alpha_f)}{OUT(p,t) + \mid C(t) \mid (1 - k\alpha_f)}$	(c_5') $\forall t \in T$ persistent or immediate : ${}^\bullet t = \{p\}$, $W^-(p,t) = f,\ A(f) = 1$ $\wedge k \in \mathbb{N} : k\alpha_f \leq B_p \leq (k+1)\alpha_f$
$x_t \bar{S}_t \geq OUT(p,t)(\bar{M}[p] + \frac{B_p}{B_q}\bar{M}[q] - B_p)$	(c_6) $\forall t \in T$ persistent or immediate: ${}^\bullet t = \{p,q\}$, $W(p,t) = f, W(q,t) = g, A(f) = A(g) = 1, \mid f \mid = \mid g \mid = 1$
$\alpha_f x_t \bar{S}_t \geq OUT(p,t)\bar{M}[p] + \mid C(t) \mid (1 - \alpha_f) +$ $-OUT(p,t)B_p \cdot \left(\mid C(t) \mid - \frac{OUT(q,t)\bar{M}[q] + \mid C(t) \mid (1-\alpha_g)}{OUT(q,t)B_q + \mid C(t) \mid (1-\alpha_g)} \right)$	(c_6') $\forall t \in T$ persistent or immediate : ${}^\bullet t = \{p,q\}$, $B_p \leq B_q,\ W(p,t) = f, W(q,t) = g, A(f) = A(g) = 1$
$\alpha_1 x_t \bar{S}_t \geq OUT(p1,t)\bar{M}[p1] - \mid C(t) \mid (-\alpha_1 + 1) +$ $-OUT(p1,t)B_{p1} \max_{1 \leq j \leq n} fj$	(c_7) $\forall t \in T$ persistent or immediate: ${}^\bullet t = \{p1, \ldots, pn\}, B_{p1} \leq B_{pj}, j \in \{2, \ldots, n\}$,
$fj = 1 - \frac{OUT(pj,t)\bar{M}[pj] + \mid C(pj) \mid (-\alpha_j + 1)}{B_{pj} / \mid C(pj) \mid - \alpha_j + 1}$	$W(pi,t) = f_i,\ A(f_1) = 1$
$\bar{M}, \vec{x}, \vec{\sigma} \geq 0$	(c_8)

TABLE 1. Linear programming problem.

The average marking equation is written here in explicit form, but it could be written also in matricial form. Moreover relation (c_7) has been derived for TWN's under the hypothesis of strong symmetries. In particular we assumed that, for each input place of transition t in inequality (c_7), the weights of the arcs belonging to the families corresponding to the function labelling the arc are the same. Obviously the uncoloured version of (c_7) has no such restriction.

As we remarked in the case of timed Petri nets, also for TWN's constraint (c_2) becomes an equality for bounded places (c_2'). The equality sign also holds true in (c_4) if $\alpha_f = 1$ (i.e. the unique family of arcs corresponding to function f have weight 1) since in this case it may be combined with the opposite inequality (c_5). For the case of places with several output conflicting transitions, inequality (10) derived in previous section (or its coloured counterpart) can be added if preselection policy is assumed for the resolution of the conflict. The constraint

labelled with (c_5) can be improved if the input place to t_i is bounded, by introducing the additional constraint (c_5').

The LPP of table 1 provides a general method to compute upper and lower bounds for arbitrary linear functions of average marking of places and throughput of transitions. For instance, if $f(\bar{M}, \vec{x}) = x_i$, then the problem can be used to compute an upper or a lower bound (depending on the selection of "max" or "min" optimization for the objective function) for the throughput of transition t_i. In an analogous way, upper or lower bounds for the average marking of a given place p_j can be derived by solving the LPP of table 1 for the objective function $f(\bar{M}, \vec{x}) = \bar{M}[p_j]$. The bounds are insensitive to the timing probability distributions since they are based only on the knowledge of the average service times.

Notice also that most equalities and inequalities contain coefficients that depend only on the net structure and on the (known) average transition firing times (and probabilities in case of free choice immediate conflicts). The only coefficients that may be unknown at the time of the formulation of the model are the actual bounds for places B_i. If the modeller has no a-priori more precise knowledge of these bounds, notice that an upper bound for them that can be used in the LPP of table 1 may be computed from a simplified LPP that contains only constraint c_1 (structural marking bound).

An improvement of the proposed bounds can be obtained if additional constraints that improve the linear characterization of the average marking in terms of the equation $\bar{M} = M_0 + C \cdot \vec{\sigma}$ are considered. For instance, if a *trap* P_T (i.e., $P_T \subseteq P, P_T^\bullet \subseteq {}^\bullet P_T$) is not a P-semiflow, the net is live, and we are interested only in the steady state performance, then we can add the constraint: $\sum_{p_k \in P_T} \bar{M}[p_k] \geq 1$.

Similarly, if a *siphon* P_S ($P_S \subseteq P$, ${}^\bullet P_S \subseteq P_S^\bullet$) is not a P-semiflow and the net is live, then we can add the constraint: $\sum_{p_k \in P_S} \bar{M}[p_k] \geq 1$.

The systematic method for the improvement of linear characterization of reachable markings based on the addition of *implicit places*, presented in [9], can be also applied as in [5].

We remark that linear programming problems can be solved in *polynomial time* [14], therefore the above presented method for the computation of (upper and lower) bounds for the throughput and for the average marking of general timed nets has a polynomial complexity on the number of nodes of the net. Moreover, the *simplex* method for the resolution of LPP's proceeds in linear time in most cases even if it has a theoretically exponential complexity.

Similar results, based on linear programming techniques, were presented in previous works [3, 4, 2] for the computation of throughput upper bounds for particular net subclasses, such as marked graphs or free choice nets. The new approach derived in this section generalizes those recent results in two ways: first, it can be applied to arbitrary Petri net structures; second, it allows one to compute upper and lower bounds for throughput and average marking in a simple and unified way. The proposed method produces the same results as previous ones [3, 4, 2] when the same net subclasses are considered.

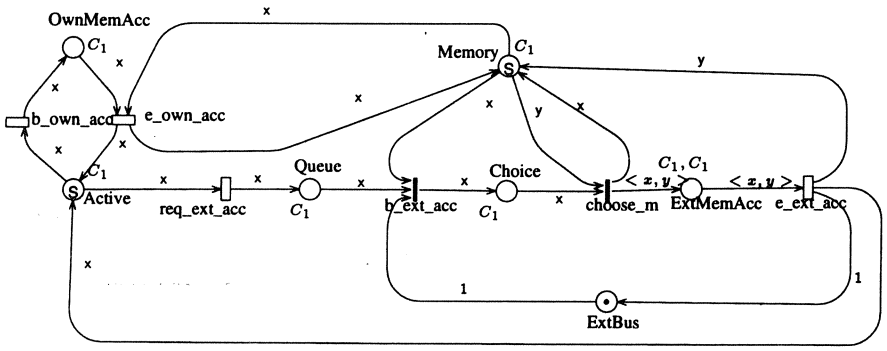


FIGURE 1. TWN model of a shared-memory multiprocessor.

4 AN EXAMPLE OF APPLICATION

Let us present an example of application for the computation of bounds in the case of the TWN of figure 1. The architecture comprises a set of processing modules interconnected by a common bus called the "external bus". A processor can access its own memory module directly from its private bus through one port, or it can access non-local shared-memory modules by means of the external bus. In case of contention for the access to one shared-memory module, preemptive priority is given to external access through the external bus with respect to the accesses from the local processor. The experiments on the shared-memory model have been carried out assuming to have 4 processors and that the average service time of all the transitions are equal to 0.5. According to the arguments presented in the previous sections, bounds can be computed solving LPP's with constraints included in table 2, where the first letters of each transition name have been used for reasons of space. The solution for the LPP leads to upper and lower bounds, for the throughput of transitions, given by $\frac{8}{11} \leq x_{e\_e\_a} \leq 2$, while the "exact" solution with exponential distribution is $x_{e\_e\_a} = 1.71999$. An improvement in the lower bound can be obtained observing that when a token arrives in place Choice transition choose_m is enabled at least for one transition instance. This implies that the average marking of place Choice is equal to 0 (transition choose_m is immediate), so $\bar{M}[Choice] = 0$ and $B_{Choice} = 0$ (only tangible markings are considered) can be added to the set of constraints. Moreover place Memory is implicit w.r.t. the enabling of transition b_ext_acc, so we can consider this transition as having only two input places, so constraint (c_6) can be applied instead of constraint (c_7). Finally $B_{Queue} = 3$ can be added since the output transition of place Queue is immediate, and from the behaviour of the model it is clear that at most 3 processors can be waiting in the queue. The relations (c_7) in the above LPP can thus be replaced with

(c_1)	$\bar{M}[Active] = 4 + \sigma_{e\_e\_a} + \sigma_{e\_o\_a} +$
	$-\sigma_{r\_e\_a} - \sigma_{b\_o\_a};$
	$\bar{M}[Memory] = 4 + \sigma_{e\_e\_a} - \sigma_{b\_e\_a};$
	$\bar{M}[OwnMemAcc] = \sigma_{b\_o\_a} - \sigma_{e\_o\_a};$
	$\bar{M}[Queue] = \sigma_{r\_e\_a} - \sigma_{b\_e\_a};$
	$\bar{M}[Choice] = \sigma_{b\_e\_a} - \sigma_{c\_m};$
	$\bar{M}[ExtMemAcc] = \sigma_{c\_m} - \sigma_{e\_e\_a};$
	$\bar{M}[ExtBus] = 1 + \sigma_{e\_e\_a} - \sigma_{b\_e\_a};$
(c_2')	$x_{e\_e\_a} + x_{e\_o\_a} = x_{r\_e\_a} + x_{b\_o\_a};$
	$x_{b\_e\_a} = x_{c\_m} = x_{e\_e\_a} = x_{r\_e\_a};$
(c_3)	$x_{b\_o\_a} = x_{r\_e\_a};$
$(c_4 \& c_5)$	$x_{b\_o\_a}\bar{S}_{b\_o\_a} = \frac{\bar{M}[Active]}{2};$
	$x_{r\_e\_a}\bar{S}_{r\_e\_a} = \frac{\bar{M}[Active]}{2};$
	$x_{e\_e\_a}\bar{S}_{e\_e\_a} = \bar{M}[ExtMemAcc];$
(c_4)	$x_{e\_o\_a}\bar{S}_{e\_o\_a} \leq \bar{M}[OwnMemAcc];$
	$x_{e\_o\_a}\bar{S}_{e\_o\_a} \leq \bar{M}[Memory];$
(c_6)	$x_{e\_o\_a}\bar{S}_{e\_o\_a} \geq \bar{M}[OwnMemAcc] +$
	$+\frac{B_{OwnMemAcc}}{B_{Memory}}\bar{M}[Memory] - B_{Memory};$
(c_7)	$4(\bar{M}[ExtBus] - B_{ExtBus}(1 - \frac{\bar{M}[Memory]}{B_{Memory}})) \leq 0$
	$4(\bar{M}[ExtBus] - B_{ExtBus}(1 - \frac{\bar{M}[Queue]}{B_{Queue}})) \leq 0;$

TABLE 2. Constraints for the model in figure 1.

the new constraint:

$$4(\bar{M}[ExtBus] + \frac{B_{ExtBus}}{B_{Queue}}\bar{M}[Queue] - B_{ExtBus}) \leq 0$$

where $B_{Queue} = 3$. Solving this reduced linear programming problem the values obtained for the upper and lower bounds are:

$$1 \leq x_{e\_e\_a} \leq 2$$

5 CONCLUSIONS

Operational analysis of timed Petri net models has been introduced. In particular, we have defined adequate observable quantities that allow the derivation of fundamental relations among them. These relations hold true "operationally," i.e., in each sample path that one can measure in an experiment. Among these relations the enabling operational law constitutes a restatement of the classical utilization law (derived in the framework of multiple server queues) for each timed transition of a general Petri net model with infinite server semantics. Bounding inequalities in both directions between throughput of a transition and average marking of its input places have also been derived. These results are typical on network models containing synchronization, and represent

a novel result of operational analysis. Under the hypothesis of strong symmetries, analogous relations have been derived for Timed Well-Formed nets.

A direct and interesting application of the obtained operational laws is the computation of performance bounds insensitive to the timing probability distributions. Indeed the bounding technique proposed in this paper guarantees that the exact value of a given performance index falls in the computed interval, whatever its probability distributions is. In this sense this bound technique is substantially more robust with respect to practical application than any performance evaluation technique based on Markovian analysis or simulation (where in any case some hypothesis on the timing distribution must be introduced in order to produce sample execution traces).

Proper linear programming problems including the derived operational laws as constraints allow one to estimate upper and lower bounds for arbitrary linear functions of the throughput and the average marking (in particular, the throughput of a single transition or the average marking of a particular place). This approach constitutes a clear improvement and generalization of previous results valid only for particular net subclasses. An important characteristics of this new method is that it is "open" to the introduction of additional constraints besides the ones already described in this paper provided that they are expressed in linear algebraic form. The straightforward addition of some constraints deriving from a specific knowledge about some peculiar behavioural characteristics of a WN model may improve the quality of the bounds based on results developed for the analysis of the qualitative behaviour of untimed Petri net models.

The proposed method for bounds computation is cheap, since the solution of the LPPs is practically extremely fast in terms of CPU time compared to Markovian numerical analysis (not to mention simulation).

The size of the LPP depends only on the net structure (number of places, transitions and arcs); in particular it is also independent of the cardinality of the basic colour classes, thus adding a dimension on the parameterization of the results. If the computation of bounds for a 4 processor system takes less than 1 second of CPU time, it will take the same order of magnitude to compute bounds for a 1,000 processor system.

References

1. M. Ajmone Marsan, G. Balbo, A. Bobbio, G. Chiola, G. Conte, and A. Cumani, "The effect of execution policies on the semantics and analysis of stochastic Petri nets," *IEEE Trans. on Soft. Eng.*, 15(7):832–846, July 1989.
2. J. Campos, G. Chiola, J.M. Colom, and M. Silva, "Properties and performance bounds for timed marked graphs," *IEEE Trans. on Circ. and Syst. I: Fundamental Th. and App.*, 39(5):386–401, May 1992.
3. J. Campos, G. Chiola, and M. Silva, "Ergodicity and throughput bounds for Petri nets with unique consistent firing count vector," *IEEE Trans. on Soft. Eng.*, 17(2):117–125, Feb. 1991.

4. J. Campos, G. Chiola, and M. Silva, "Properties and performance bounds for closed free choice synchronized monoclass queueing networks," *IEEE Trans. on Aut. Cont.*, 36(12):1368–1382, Dec. 1991.
5. J. Campos, J.M. Colom, and M. Silva, "Improving throughput upper bounds for net based models," In *Proc. of the IMACS-IFAC Symp. Modelling and Control of Tech. Syst.*, pp. 573–582, Lille, May 1991.
6. G. Chiola, "A graphical Petri net tool for performance analysis," In *Proc. of the* 3^{rd} *Intern. Workshop on Modeling Techniques and Performance Evaluation*, Paris, March 1987.
7. G. Chiola and C. Anglano, "Linear programming performance bounds for symmetric coloured nets," *Tech. Rep.*, Dip. di Informatica, Univ. di Torino, Feb. 1993.
8. G. Chiola, C. Dutheillet, G. Franceschinis and S. Haddad, "Stochastic Well-Formed Coloured nets for symmetric modelling applications," *IEEE Trans. on Comp.*, 42:1343–1360, 1993.
9. J.M. Colom and M. Silva, "Improving the linearly based characterization of P/T nets," In G. Rozenberg, ed. *Advances in Petri Nets 1990*, Vol. 483 of *LNCS*, pp. 113–145. Springer-Verlag, Berlin, 1991.
10. Y. Dallery and X.R. Cao, "Operational analysis of stochastic closed queueing networks," *Performance Evaluation*, 14:43–61, 1992.
11. P.J. Denning and J.P. Buzen, "The operational analysis of queueing network models," *ACM Computing Surveys*, 10:225–262, 1978.
12. E. Gelenbe, "Stationary deterministic flows in discrete systems: I," *Theoretical Computer Science*, 3(2):107–127, April 1983.
13. T. Murata, "Petri nets: Properties, analysis, and applications," *Proceedings of the IEEE*, 77(4):541–580, April 1989.
14. G.L. Nemhauser, A.H.G. Rinnooy Kan, and M.J. Todd, eds. *Optimization*, Vol. 1 of *Handbooks in Operations Research and Management Science*. North-Holland, Amsterdam, 1989.
15. M. Silva, "Introducing Petri Nets," Chapter 1 of *Practice of Petri Nets in Manufacturing* (F. Dicesare et al.). Chapman & Hall, 1993.

Computing Bounds for the Performance Indices of Quasi-Lumpable Stochastic Well-Formed Nets

Giuliana Franceschinis *
Università di Torino
Dipartimento di Informatica

Richard R. Muntz†
University of California at Los Angeles
Computer Science Department

Structural symmetries in Stochastic Well-Formed Colored Petri Nets (SWN) lead to behavioral symmetries that can be exploited using the Symbolic Reachability Graph (SRG) construction algorithm: it allows to compute an aggregated Reachability Graph (RG) and a "lumped" Continuous Time Markov Chain (CTMC) that contain all the information needed to study the qualitative properties and the performance of the modeled system respectively. Some models exhibit qualitative behavioral symmetries that are not completely reflected at the CTMC level, we call them *quasi-lumpable* SWN models. In these cases, exact performance indices can be obtained by avoiding the aggregation of those markings that are qualitatively but not quantitatively equivalent. An alternative approach consists of aggregating all the qualitatively equivalent states, and computing approximated performance indices. In this paper a technique is proposed to compute bounds on the performance of SWN models of this kind, using the results presented in [4]. The technique is based on the Courtois and Semal's bounded aggregation method [2].

1 Introduction

The Coloured Petri Net (CPN) formalism [7] was initially introduced for model design convenience: indeed more compact and parametric models can be built using CPN instead of the classic PN. Structural symmetries in CPN models,

*The work of G. Franceschinis was done while she was visiting UCLA CSD with the support of the CNR "Progetto Finalizzato Sistemi Informatici e Calcolo Parallelo", and of the Italian MURST "40%" Project.

†The work of Richard Muntz was supported under NSF grant CCR 9215064.

often lead to behavioral symmetries that can be exploited to reduce the computational cost of analysis methods based on the reachability graph construction [6, 5]. When stochastic (exponential) timing is associated to transitions, the modeled system performance can be computed by analyzing the continuous time Markov chain (CTMC) isomorphic to the model reachability graph, hence in case behavioral symmetries apply also at the CTMC level, they can be used to lower the cost of performance analysis as well [3, 9].

The Stochastic Well-Formed Colored Petri Net (SWN) formalism [1] has been introduced to systematize the symmetry exploitation technique: behavioral symmetries of models described with this formalism can be automatically discovered and exploited by defining equivalence classes of markings called *symbolic markings*. A *Symbolic Reachability Graph* (SRG) can be directly generated from a SWN; it retains enough information to study the qualitative properties of the model and to derive a *lumped* CTMC from which all the desired performance indices can be computed. The core of the SRG generation method relies on the definition of object types, the basic color classes, and of their partition into a number of disjoint subsets of homogeneously behaving objects. Equivalence classes of markings are defined as sets of markings that are equal up to a permutation of homogeneously behaving objects. As a consequence of this equivalence relation definition, the potential aggregation decreases as the partitioning of classes into subclasses increases. As we'll see, it may happen that the partitioning of classes into several subclasses is needed only for the correct specification of the *quantitative* behavior of the model (i.e., the transition rates), while they're not used to describe the qualitative behavior (i.e., the token flow in the net). Hence the subclasses used only for the definition of transition rates could be merged when performing a qualitative analysis. By merging them also in the performance analysis phase an error is introduced, and approximated performance results are obtained. In this paper we propose a technique that allows to obtain bounds (rather than just approximations) on the performance indices of SWN models when the states aggregation is performed according to qualitative behavioral symmetries that are not completely reflected at the CTMC level. The bound computation algorithm used, is the one proposed in [4]. The contribution of this paper is a method for defining the *state aggregates* for the application of the above bounding method. Actually the presented result is stronger since the proposed method also allows to directly compute the aggregated MC that is actually solved to compute the bounds, without ever computing the complete quasi-lumpable chain. The method can thus be considered as an extension of the SRG exact aggregation method.

The paper is organized as follows: in Section 2 the SWN formalism and the associated RG aggregation techniques are informally defined; in Section 3 the method for computing bounds of quasi-lumpable MC is summarized; in Section 4 the main contribution of this paper is described: it comprises a structural analysis algorithm to check whether a given SWN is quasi-lumpable, and a modified SRG generation algorithm for the generation of the aggregate MC used for the bounds computation. In Section 5 two application examples are shown. Finally in Section 6 we draw some conclusions and discuss the possible

future developments of the method.

2 Stochastic Well-Formed Colored Petri Nets: an Informal Introduction

In this section the SWN formalism and the Symbolic Reachability Graph generation technique are informally introduced by means of two examples.

Let's consider a polling system comprising a set of waiting rooms (of limited capacity) where customers can queue up, and a set of servers that cyclically visit the waiting rooms to serve the customers. A GSPN model of such a system can be built up by properly linking several subnets: one subnet for each waiting room and one subnet to represent the servers behavior. If the system is homogeneous, i.e., if the system components of the same type behave similarly, a more compact representation can be obtained by explicitly modeling only one component of each type, and adding some annotation to specify how many instances of the submodel are present in the whole net, and how they are connected.

The SWN model of the polling system in Figure 1 implements this idea: the net represents the possible states of a generic queue and of a generic server: tokens in place *Thinking* represent customers still out of the waiting room, tokens in place *Waiting* represent customers queued up in the waiting room, tokens in place *busy_servers* represent customers being served, tokens in place *pos_server* represent idle servers looking for some customer to be served, while tokens in place *serv_out* represent a server moving from a queue to the next one. Transitions represent the possible state changes.

The tokens contained into places are no more undistinguishable, they carry some information needed to distinguish either customers associated with different queues or different servers. The type of data associated with tokens in a given place - the *place color domain* - can be structured, i.e., it can comprise several "fields", each with an associated basic data type. The basic data types are finite and non empty sets called *basic color classes*; in the polling system example there are two basic color classes: $Servers = \{w_1, \ldots, w_n\}$ and $Queues = \{q_1, \ldots, q_m\}$. The color domain of place *busy servers* is structured: it comprises a type *Queues* field and a type *Servers* field representing the customer being served and the server performing the service respectively. Hence the tokens contained in this place are pairs $\langle q_i, w_j \rangle$.

The basic color classes may be *ordered* (in this case a successor function must be defined on the class) and may be partitioned into several *static subclasses*, each containing homogeneously behaving objects. In our example, if the servers visit the queues in cyclic order, the class of queues could be ordered according to the visit sequence: the successor function on the *Queues* class could be defined as $\oplus\ q_i = q_{(i+1) \bmod m}$. If some queues must be served by at most one server at a time, while others can be served by many servers concurrently, then the corresponding class could be partitioned into two subclasses: $Queues = OneSrvQ \cup MltSrvQ$. If two types of servers exist that have different service speed, class *Servers* should be partitioned into two subclasses: $Servers =$

$FastSrv \cup SlowSrv$. Observe that the *Queues* class partitioning is used to distinguish objects with different *qualitative* behavior, while the *Servers* class partitioning is used to distinguish objects with same qualitative but different *quantitative* behavior.

A transition in a SWN model actually represents several transitions in the corresponding "unfolded" GSPN model: for example transition *end_serv* represents a generic end of service happening at any queue. In SWNs transitions are parameterized: the parameter types are chosen among the set of basic color classes. Transition *end_serv* has two parameters, *srv* and *queue*, of type *Servers* and *Queues* respectively. A *transition instance* is obtained assigning actual objects of proper type to the transition parameters; enabling and firing is defined only for transition instances. We denote $[t, c]$ an instance of t, where c is a tuple belonging to the transition color domain whose elements are the objects assigned to the transition parameters. The set of "colored" tokens that are withdrawn from/added to the input/output places of a given transition when one of its enabled instances fires, are defined through *arc expressions* labelling the net arcs.

The enabling of a transition instance depends both on the multiset of colored tokens contained in its input and inhibitor places and on the evaluation of an optional *predicate* associated with the transition. In the polling system model, predicates are associated with three transitions: $start\_srv_1$, $start\_srv_2$ and $Walk$. The predicate $[d(queue) = OneSrvQ]$ ($[d(queue) = MltSrvQ]$) indicates that the instances of $start\_srv_1$ ($start\_srv_2$) that may be enabled are only those with an object from subclass $OneSrvQ$ ($MltSrvQ$) assigned to parameter *queue*. Indeed this transition represents the start of a service for a customer in a queue allowing only one (multiple) service at a time. Finally, the predicate $[nextq = \oplus queue]$ indicates that in order for a $Walk$ instance to be enabled, the parameter *nextq* must be assigned the successor ($\oplus$) of the object assigned to parameter *queue*; this predicate is used to model the cyclic path followed by the servers.

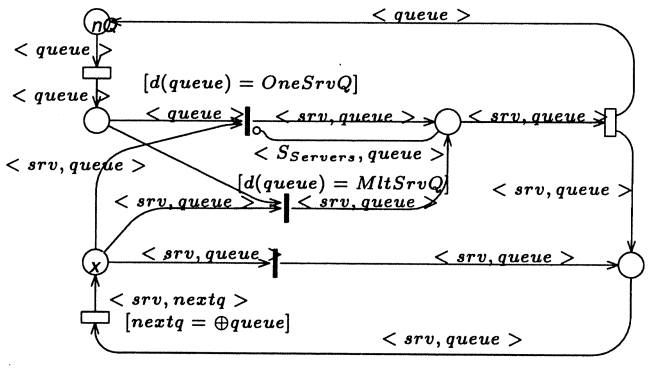


FIGURE 1. SWN model of a polling system

There are two more functions to be defined to complete the SWN definition:

the transition priority and weight functions (denoted π_t and θ_t respectively). Transition instances may have different priority levels, the enabling rule must be modified to take into account priority.

The transition weight function θ_t defines the firing rates of transition instances and is used to solve probabilistically the conflicts that may arise during the net evolution. The transition rates may depend on the particular transition instance, however there are constraints on the kind of dependence allowed, necessary to guarantee that all the objects belonging to the same static subclass behave homogeneously *by construction.*

The Symbolic Reachability Graph A major interest of SWNs is that they provide a modeling framework in which symmetries appear naturally as a way of reducing both the complexity of the representation and the state space explosion problem [1]. Symmetries in SWN are implicitly defined at the color class level, by means of *symmetry functions.* A symmetry function s_i on an ordered/unordered color class C_i is any rotation/permutation of the objects in the class, preserving static subclass partition. A symmetry function s applicable to place markings and transition color instances is a family of symmetry functions on color classes: $s = \{s_1, s_2, \ldots, s_n\}$. We denote ξ the set of all such functions.

The marking equivalence classes, called *symbolic markings*, are defined as follows:

DEFINITION 1 (SYMBOLIC MARKING) *Let Eq be the equivalence relation defined by:*

$$M \; Eq \; M' \iff \exists s \in \xi, \; M' = s.M$$

An equivalence class of Eq is called a symbolic marking, denoted with $\mathcal{M}$.

In [1] some propositions can be found, stating that the possible future qualitative and quantitative evolution of the model is the same for all the markings in an equivalence class.

Let us illustrate the above concepts through an example. Consider a closed queueing system composed of two service centers in tandem; let's assume that there are 5 customers in the system. Customers cycle between the two service centers. The first one is a single server machine with an associated exponentially distributed random delay with parameter μ. The second is a four servers machine, and each server S_i has the same exponentially distributed delay with parameter λ. A customer chooses randomly one of the four servers on each visit to the multiserver node; the same probability is associated with each server. In Figure 2 a SWN representation of the system is depicted.

Only one color class is used in this model, namely $Lines = \{l_1, \ldots, l_4\}$, representing the four servers in the multiserver queue. The presence of n tokens with associated color l_i in place $Line1$ means that n customers are queued up to get service from server S_i. A possible marking of this model is $M = Line1(\langle l_1 \rangle, \langle l_2 \rangle, \langle l_3 \rangle, 2\langle l_4 \rangle)$. . There are three markings equivalent to M,

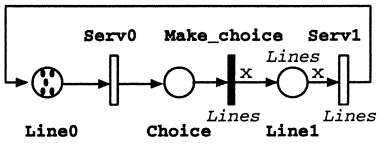


FIGURE 2. A SWN model of a queueing system

that can be obtained by applying all the possible *Lines* objects permutations to M. Thus M belongs to a symbolic marking $\mathcal{M}$ of cardinality four, that could be represented as follows: $\mathcal{M} = Line1(\langle z_1 \rangle, \langle z_2 \rangle, \langle z_3 \rangle, 2\langle z_4 \rangle)$, where z_i are variables representing objects in *Lines* and all the ordinary markings belonging to $\mathcal{M}$ can be obtained by assigning actual objects to the variables. Four transition instances are enabled in all the markings belonging to $\mathcal{M}$: $[Serv1, l_i], i = 1, \ldots, 4$, all with the same rate λ. It is possible to relate the arcs

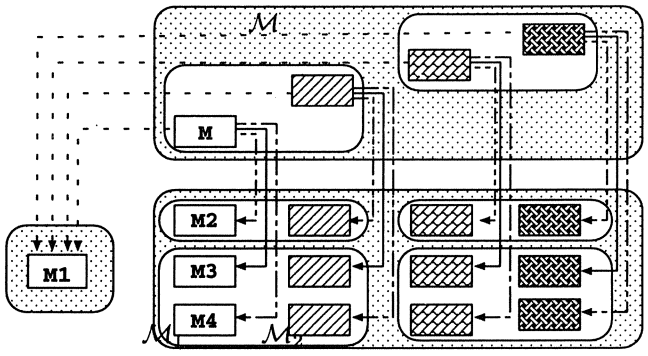


FIGURE 3. Difference in state aggregation when changing the color class partitioning.

exiting from pairs of equivalent markings: in Figure 3 related arcs are represented with similar dashed lines (equivalence classes are represented by dotted boxes). In the same figure it is possible to observe that from the symbolic marking $\mathcal{M}$, two symbolic markings can be reached: $\mathcal{M}_1$ (containing only one ordinary marking) and $\mathcal{M}_2$ (containing twelve ordinary markings). Three out of four firing instances exiting from M end up in the same symbolic marking $\mathcal{M}_2$. Actually it is possible to know in advance which firing instances lead to markings in the same equivalence class since all the objects that have the same *distribution of tokens* in places can be interchanged in a firing instance without changing the reached symbolic marking, this is the case of objects l_1, l_2 and l_3 in marking M. In order to exploit this property, it is convenient to use a representation for symbolic markings that keeps objects with the same distribution of tokens grouped into sets. We call these sets *dynamic subclasses* and denote them Z^j_{class}. The new representation of $\mathcal{M}$ using dynamic subclasses is: $\mathcal{M} = Line1(\langle Z^1_{Lines} \rangle, 2\langle Z^2_{Lines} \rangle)$ where $|Z^1_{Lines}| = 3$ and $|Z^2_{Lines}| = 1$. The *symbolic firing* instances enabled in $\mathcal{M}$ are $[Serv1, Z^1_{Lines}]$ and $[Serv1, Z^2_{Lines}]$, the former actually represents three ordinary firing instances.

If the servers in the multiserver queue had not the same service rates, for example if two servers were slower (rate λ_1) while the other two were faster (rate $\lambda_2 > \lambda_1$), then we should have partitioned class *Lines* into two static subclasses, $Lines_i, i = 1, 2$, each containing two servers with service rate λ_i. This partition causes a splitting of some symbolic marking as shown in Figure 3 (equivalence classes are represented by white boxes). Symbolic marking $\mathcal{M}$ is now split into two symbolic markings: this is needed to distinguish the case where the server with two customers in its queue is a slow one from the case in which that server is a fast one. Similarly $\mathcal{M}_2$ is now split into four symbolic markings.

Observe that in this case qualitative analysis can be performed on the more compact SRG, while exact performance results can be obtained only from the larger SRG. In the following sections we'll show a method to compute bounds for the performance indices, working with a MC of size equal to that of the more compact SRG.

3 Bounds for Quasi-Lumpable Markov Chains

In this section the method presented in [4] to compute bounds for *quasi-lumpable* Markov chains is summarized. It is based on the bounds computation method proposed by Courtois and Semal in [2]. The main result from [2] we use, is a theorem stating that it is possible to compute upper and lower bounds for the steady state probability vector of a DTMC P, when only a (componentwise) lower bound $P^- \leq P$ of the transition probability matrix is available.

In [2] two applications of the above theorem are presented: (1) computation of bounds on conditional steady state probability of a subset of states $\mathcal{S}'$ in a DTMC when the transition probability among the states in the subset is known while only partial or no information is available about the transition probability between states in $S - \mathcal{S}'$ and states in $\mathcal{S}'$; (2) computation of bounds on the steady state probability vector of a large system by decomposition into smaller subsystems: this is called the *bounded aggregation method.* This method consists of two steps: (a) computation of bounds on the conditional state probabilities within each aggregate using the method just explained; (b) computation of bounds on the probability of being in each aggregate (this requires the derivation of a lower bound $\mathcal{P}^-$ for the inter-aggregate probability matrix $\mathcal{P}$ using the results of the previous step).

Finally the results obtained in the two steps can be combined to compute bounds on the steady state probability of the original states.

The proposed method The method proposed in [4] allows to compute bounds for quasi-lumpable Markov chains. We first give a definition of quasi-lumpable MC, then intuitively describe the bounding method.

Definition 2 *A Continuous Time Markov Chain is said to be* ϵ-quasi-lumpable *with respect to a given state space partition* $\mathcal{A}$ *if its infinitesimal generator* Q

can be rewritten as $Q = Q^- + Q^\epsilon$ *, where* Q^- *is a maximal lower bound (componentwise) for* Q *that satisfies the strong lumpability conditions [8] with respect to* $\mathcal{A}$*, and no element in* Q^ϵ *is greater than* ϵ *in value.*

The intention is that Q^ϵ is a matrix with many more zero elements than Q^- and with relatively small non-zero elements. Henceforth we use the term "quasi-lumpable CTMC" for a CTMC that is ϵ-quasi-lumpable for some ϵ.

The CTMC corresponding to the RG of the SWN model of Figure 2 when the servers in the multiserver station do not have all the same rate, is ϵ-quasi-lumpable with respect to the aggregation induced by the SRG generation algorithm when *Lines* is not partitioned into static subclasses; in this case ϵ is proportional to the difference $\lambda_2 - \lambda_1$.

The bounds computation method we have proposed in [4], can be described as a simplified version of the bounded aggregation method. Actually we apply only the second step of the bounded aggregation algorithm, because our aim is to deal directly with the lumped process. The lower bound $\mathcal{Q}^-$ for the aggregate matrix $\mathcal{Q}$ is obtained by computing the minimum row sum in each submatrix $Q_{i,j}$ of transition rates from states in the i^{th} aggregate to states in the j^{th} aggregate; from matrix $\mathcal{Q}^-$ lower and upper bounds on the aggregates steady state probability can be computed by means of the Courtois and Semal method. As we'll see later, matrix $\mathcal{Q}^-$ can be computed without ever computing the complete matrix Q. Observe that it is also possible to compute an upper bound $\mathcal{Q}^+$ for the transition rates between aggregates, by taking the maximum among the row sums of each submatrix. Using the following theorem it is possible to exploit the knowledge of an upper bound $\mathcal{Q}^+$ to get improved bounds:

THEOREM 1 *[4] Let* Q *be the infinitesimal generator of an ergodic CTMC with* n *states. Let* Q^- *be a lower bound (componentwise) for* Q*, i.e.,* $Q^- \leq Q$*. Let* $\mathbf{y}^T = (Q - Q^-)\mathbf{e}^T$ *and finally let* $Q_s = Q^- + \mathbf{y}^T\mathbf{x}$ *where* $\mathbf{x}$ *is an unknown row vector such that* $\mathbf{x}\mathbf{e}^T = 1$*. There exists a vector* $\mathbf{x}$ *such that the steady state probability of* Q_s *is equal to that of* Q*.*

Proof: The theorem is proven by showing that $x = \frac{\pi(Q-Q^-)}{\pi(Q-Q^-)\mathbf{e}^T}$ is a vector that satisfies the required property. The complete proof can be found in [4]. □

The bounds improvement method can be applied after a first set of bounds π^- and π^+ have been computed by means of the Courtois and Semal's method.

4 AUTOMATIC DERIVATION OF THE AGGREGATED MC FOR BOUNDS COMPUTATION

In order for the bounding method to be convenient (from a computational cost point of view), the ideal situation would be to derive the lumped matrices $\mathcal{Q}^-$ and $\mathcal{Q}^+$ directly from the high level model without having to compute (or at least to store) the large complete matrix. In this section we present a method that allows to reach this goal for a subclass of SWN models called *quasi-lumpable* SWNs.

The issue of automatic construction of the lumped matrix is closely related to that of the detection of symmetries on the state space that induce a partition

into state aggregates. The same problem has to be faced when exact lumping methods are used, the above requirement of constructing automatically the aggregated CTMC for bounds computation, is similar to that of directly obtaining the lumped CTMC from the SRG. We now show that the same approach used to derive the lumped CTMC from the SRG can be extended to our bounding method, that is the upper and lower bound aggregated matrices $\mathcal{Q}^-$ and $\mathcal{Q}^+$ can be automatically built using a modified version of the *SRG* algorithm. The bounding matrices can be computed at different accuracy levels; as usual higher accuracy can be achieved with higher computational cost.

In Section 2 we introduced the concept of color classes and subclasses in SWN. The color classes are used to define sets of "similar" objects and the partition of a class into static subclasses is used to identify subsets of objects in a class that share the same behavior. As already pointed out, it is possible to distinguish between two possible situations: (1) the objects in different subclasses have different *qualitative* behavior, i.e., they cannot play the same "role" in the system because they have different possible evolutions, (2) the objects in different subclasses have the same *qualitative* behavior, but the event sequences happen with different rates depending on which subclass the object belongs to. For example, in the model of Figure 2 the colored tokens representing customers in the "multiserver" station follow the same route through transition *Serv*1, independently of their color, however, the firing rate of the transition depends on the token color.

Hence, given a SWN model of the system under study, a first analysis is needed to detect all the subclasses in each class with similar "qualitative" behavior. This permits automatic determination of the candidate quasi-lumpable state aggregates. Then two approaches are possible, (1) apply the usual Symbolic Reachability Graph generation algorithm to the system as it is specified and use the information about aggregate states to compute the lumped matrices $\mathcal{Q}^-$ and $\mathcal{Q}^+$ in a second step; (2) apply the SRG generation algorithm to a modified model where some of the homogeneously behaving subclasses are merged, and apply a modified CTMC transition rate computation rule to directly derive $\mathcal{Q}^-$ and $\mathcal{Q}^+$ from the SRG.

In the sequel we present the subclasses merge algorithm and the modified MC transition rate computation rule.

Subclasses merge algorithm For each basic color class, the following static classes merge procedure has to be performed:

old_static_i = ⟨set of static subclasses in the original C_i⟩
$new\_static_i = \emptyset$
while $old\_static_i \neq \emptyset$ **do**
    ⟨ remove a subclass sc from old_static_i⟩
    $lsc = sc$
    $sc\_list = emptylist$
    append(sc, sc_list)
    for each $sc' \in old\_static_i$ **do**
        if $semilumpable(sc', sc\_list)$ **then**

$lsc = lsc \cup sc'$
$\langle$ remove sc' from $old\_static_i \rangle$
append(sc', sc_list)
end if
end for
$old\_sc\_list[lsc] = sc\_list$
$\langle$ add static subclass lsc to $new\_static_i \rangle$
end while

A set of static subclasses old_static_i is the input for this procedure. The output is a new set new_static_i, of static subclasses. The cardinality of the new set is less than or equal to that of the old set. For each new static subclass lsc, a list $old\_sc\_list[lsc]$ of the old static subclasses that have been merged into lsc is maintained.

Function $semilumpable(sc', sc\_list)$ implements the most important part of the algorithm. Let ξ' be the set of symmetry functions that satisfy the constraint of allowing only permutation of objects within the same static subclass for all classes $C_j \neq C_i$, while allowing the exchange of objects in $\{sc'\} \cup sc\_list$; function *semilumpable* is meant to return true iff for any $s \in \xi'$, every predicate Φ occurring in the model satisfies the equation $\Phi(c) = \Phi(s.c)$ and every arc function f occurring in the model satisfies $f(s.c) = s.f(c)$. This is surely true when no references to the subclasses in $sc' \cup sc\_list$ occur in any arc function and predicate of the model, and predicates do not contain clauses like $[d(x) = d(y)]$ (where x and y are transition parameters of type C_i). More complex sufficient conditions may be defined, for example predicates Φ/function elements f_i containing occurrences of the subclasses to be merged may be allowed if they have the form $\Phi = (\bigvee_{\forall sbc \in sc' \cup sc\_list}[d(x) = sbc]) \wedge \Phi'$ and $f_i = \alpha \sum_{sbc \in\ sc' \cup sc\_list} S_{sbc} + f_i'$ respectively, and the subclasses to be merged do not occur in Φ' or in f_i', furthermore Φ' does not contain clauses like $[d(x) = d(y)]$.

Modified MC transition rates computation rule The aggregate matrices $\mathcal{Q}^-$ and $\mathcal{Q}^+$ are computed by applying the SRG generation algorithm to a modified model where static subclasses containing objects with the same qualitative behavior but different associated rates are merged.

In a SWN, the transition firing rate of a generic transition t is defined as a function θ_t from tuples of static color classes to reals. For example transition

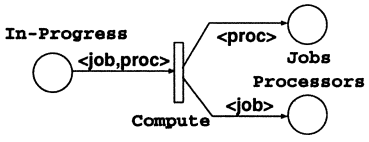


FIGURE 4. A simple SWN transition example.

Compute in Figure 4 takes a token with its two component color from its input place: the first component represents a processor and the second represents a job. If the jobs class J is divided into two subclasses of short jobs J_s and long

jobs J_l and the processors are divided into two subclasses of fast processors P_f and slow processors P_s, then a possible definition for the transition rate could be: $\theta_t(\langle < P_s, J_s > \rangle) = 0.3$, $\theta_t(\langle < P_f, J_s > \rangle) = 1$, $\theta_t(\langle < P_s, J_l > \rangle) = 0.15$, $\theta_t(\langle < P_f, J_l > \rangle) = 0.5$.

When a symbolic transition instance $[t, z]$ is fired, its rate $\lambda([t, z])$ is computed as follows: (1) derive from z the tuple d of corresponding static subclasses (each dynamic subclass is associated with exactly one static subclass, see [1]); (2) $\lambda([t, z]) = m\ \theta_t(d)$ where m is a factor that depends on the cardinality of both the subclasses in z and in d. The reason for the multiplicative factor is that a symbolic firing instance is an aggregation of m ordinary firing instances all with the same rate $\theta_t(d)$. In our example a possible symbolic instance for transition *Compute* could be $[Compute, \langle Z_P^1, Z_J^2 \rangle]$ with $|Z_P^1| = 2, Z_P^1 \in P_s$, and $|Z_J^2| = 1, Z_J^2 \in J_l$ meaning that there are 2 $(= |Z_P^1|)$ slow processors each of which is processing 1 $(= |Z_J^2|)$ long job. This symbolic instance stands for 2 "ordinary" instances that are grouped in the lumping process.

In the following we show how to derive the modified rate for a firing instance that involves objects belonging to some merged static subclass. We denote $D_{i,j}$ the new static subclasses and $\{D_{i,j}^l\}$ the set of the original subclasses that are aggregated into $D_{i,j}$. The rates of the resulting new transition instances, in general will depend on the cardinality of the dynamic subclasses in the color instance. Let $[t, z]$ be the symbolic transition instance for which a rate has to be computed. Let d be the associated static subclasses tuple. Let d' be the subtuple of d composed of merged static subclasses and z' the corresponding subtuple of z. For each element $d_k = D_{i,j}$ in d', compute the set $\mathcal{Z}_k$ of possible partitions of dynamic subclass z_k into one or more new dynamic subclasses, each associated with a different static subclass $D_{i,j}^l$ of $D_{i,j}$ (see Figure 5). The Cartesian product of the $\mathcal{Z}_k$s leads to a set of sums of original transition

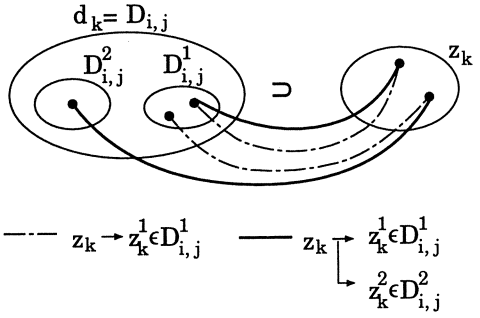


FIGURE 5. Possible partitions of z_k with respect to the "merged" static subclass $D_{i,j}$.

instances. From each sum a rate can be computed that corresponds to a value for a row sum in the aggregate transition matrix corresponding to $[t, z]$. The minimum and maximum in this set of rates gives the value for the corresponding transition rate in the aggregate matrices.

In the example of Figure 4 assume that $|P_s| = |P_f| = 2$ and $|J_s| = 2, |J_l| = 1$,

and suppose we want to merge slow and fast processors into a unique class P and short and long jobs into a unique class J. Hence the symbolic transition instance $[Compute, \langle Z_P^1, Z_J^2 \rangle]$ with $|Z_P^1| = 2, Z_P^1 \in P$ and $|Z_J^2| = 2, Z_J^2 \in J$ incorporates the following six possible instances with respect to the old classes partition.

- $[Compute, \langle Z_P^{1a}, Z_J^{1a} \rangle] + [Compute, \langle Z_P^{1a}, Z_J^{1b} \rangle]$ with $|Z_P^{1a}| = 2$ and $Z_P^{1a} \in P_s$, $|Z_J^{1a}| = 1$ and $Z_J^{1a} \in J_s$, $|Z_J^{1b}| = 1$ and $Z_J^{1b} \in J_l$; $rate = 2 \cdot 0.3 + 2 \cdot 0.15 = 0.9$
- $[Compute, \langle Z_P^{1a}, Z_J^{1a} \rangle] + [Compute, \langle Z_P^{1a}, Z_J^{1b} \rangle]$ with $|Z_P^{1a}| = 2$ and $Z_P^{1a} \in P_f$, $|Z_J^{1a}| = 1$ and $Z_J^{1a} \in J_s$, $|Z_J^{1b}| = 1$ and $Z_J^{1b} \in J_l$; $rate = 2 \cdot 1 + 2 \cdot 0.5 = 3$
- $[Compute, \langle Z_P^{1a}, Z_J^{1a} \rangle]$ with $|Z_P^{1a}| = 2$ and $Z_P^{1a} \in P_s$, $|Z_J^{1a}| = 2$ and $Z_J^{1a} \in J_s$; $rate = 4 \cdot 0.3 = 1.2$
- $[Compute, \langle Z_P^{1a}, Z_J^{1a} \rangle]$ with $|Z_P^{1a}| = 2$ and $Z_P^{1a} \in P_f$, $|Z_J^{1a}| = 2$ and $Z_J^{1a} \in J_s$; $rate = 4 \cdot 1 = 4$
- $[Compute, \langle Z_P^{1a}, Z_J^{1a} \rangle] + [Compute, \langle Z_P^{1a}, Z_J^{1b} \rangle] + [Compute, \langle Z_P^{1b}, Z_J^{1a} \rangle] + [Compute, \langle Z_P^{1b}, Z_J^{1b} \rangle]$ with $|Z_P^{1a}| = 1$ and $Z_P^{1a} \in P_s$, $|Z_P^{1b}| = 1$ and $Z_P^{1a} \in P_f$, $|Z_J^{1a}| = 1$ and $Z_J^{1a} \in J_s$, $|Z_J^{1b}| = 1$ and $Z_J^{1b} \in J_l$; $rate = 0.3 + 0.15 + 1 + 0.5 = 1.95$
- $[Compute, \langle Z_P^{1a}, Z_J^{1a} \rangle] + [Compute, \langle Z_P^{1b}, Z_J^{1a} \rangle]$ with $|Z_P^{1a}| = 1$ and $Z_P^{1a} \in P_s$, $|Z_P^{1b}| = 1$ and $Z_P^{1a} \in P_f$, $|Z_J^{1a}| = 2$ and $Z_J^{1b} \in J_s$; $rate = 2 \cdot 0.3 + 2 \cdot 1 = 2.6$

There is a computationally less expensive but not always accurate method for computing lower/upper bounds for the elements of $\mathcal{Q}^-$ and $\mathcal{Q}^+$. Given a transition t and a tuple d of aggregate static subclasses, compute the Cartesian product D of the sets of original subclasses in the component aggregate subclasses. Associate with d the $min(max)_{d' \in D} \theta_t(d')$. In this way the computed $\mathcal{Q}^-$ and $\mathcal{Q}^+$ are bounds for $\mathcal{Q}$, but they can be very inaccurate in some cases. In the previous example this simplified method would have predicted correctly the maximum rate (= 4), while the minimum (= 0.6) would have been less than the correct one (0.9).

Observe that even if the computation of the aggregate transition rates has a certain cost, it could be performed once and for all in a way that the result is easily reusable for models differing only in the actual parameter values. Moreover the computation has to be done only for the transition instances involving merged static subclasses: if the model has many transitions that do not contain the merged color class in their color domain definition there are good chances that the complex rates computation method has to be applied only to a minority of transition instances and that the corresponding overhead be compensated by the computation saving due to the stronger state aggregation.

5 Two Examples

In this section the proposed method is applied to two example models: the first model represents the two servers in the tandem system introduced in Section 2, while the second model represents a simple concurrent program mapped onto a parallel architecture.

5.1 Two servers in tandem

The model of Figure 2 is a good candidate for the application of the method. In fact this is a typical case in which the partition of the servers in two subclasses with equal speed within each subclass leads to a quasi-lumpable MC structure. We assume that the rates λ_1 and λ_2 associated with the two classes of servers are $\lambda_1 = 1.00$ and $\lambda_2 = 1.01$. Observe that in this case it is trivial to find mergeable static subclasses since neither functions of type $S_{subclass}$ appear on any arc nor predicates involving any static subclass are used.

Let's consider some transition instances to see how the lower bound matrix $\mathcal{Q}^-$ has been computed. There are two colored transitions that may be instanced to objects of *Lines*, namely *Make_choice* and *Serv*1. We describe the transition rate bounds computation for transition *Serv*1 only, a similar argument applies to *Make_choice*. Let's consider the generic symbolic instance $[Serv1, Z^j_{Lines}]$. We have to consider four cases: $|Z^j_{Lines}| = k,\ k = 1, \ldots, 4$.

i) $|Z^j_{Lines}| = 1$: There are only two possibilities to take into account: $Z^j_{Lines} \in Lines1$ and $Z^j_{Lines} \in Lines2$. Hence $min\_rate([Serv1, Z^j_{Lines}]) = \lambda_1$ and $max\_rate([Serv1, Z^j_{Lines}]) = \lambda_2$.

ii) $|Z^j_{Lines}| = 2$: There are three possibilities to consider: (1) both objects in Z^j_{Lines} belong to *Lines*1, (2) both objects in Z^j_{Lines} belong to *Lines*2 or (3) one object belongs to *Lines*1, while the other one belongs to *Lines*2. Thus we have $min\_rate([Serv1, Z^j_{Lines}]) = min(2\lambda_1, 2\lambda_2, \lambda_1 + \lambda_2) = 2\lambda_1$ while $max\_rate([Serv1, Z^j_{Lines}]) = 2\lambda_2$.

iii) $|Z^j_{Lines}| = 3$: There are two possibilities to consider: two out of three objects belong to *Lines*1 and the remaining object belongs to *Lines*2 or vice versa. Hence $min\_rate([Serv1, Z^j_{Lines}]) = min(\lambda_1 + 2\lambda_2, \lambda_2 + 2\lambda_1) = \lambda_2 + 2\lambda_1$ while $max\_rate([Serv1, Z^j_{Lines}]) = \lambda_1 + 2\lambda_2$.

iv) $|Z^j_{Lines}| = 4$: There is only one possible assignment of objects to static subclasses in this case: two objects belong to *Lines*1 while the other two belong to *Line*2 so that $min\_rate([Serv1, Z^j_{Lines}]) = max\_rate([Serv1, Z^j_{Lines}]) = 2\lambda_1 + 2\lambda_2$.

The bounds are obtained solving 17 aggregated CTMCs each comprising 18 states. The following table contains mean queue length (MQL), and throughput (THRU) bounds as well as the corresponding exact values.

	Lower bound	Exact value	Upper bound	Spread %
MQL	1.250705	1.273430	1.385868	11.0
THRU	0.967612	0.98284	0.983502	1.6

For this model it has been possible to get better bounds by applying the bounds improvement method cited in Section 3:

	Lower bound	Exact value	Upper bound	Spread %
MQL	1.250705	1.273430	1.295075	3.5
THRU	0.980303	0.98284	0.983502	0.3

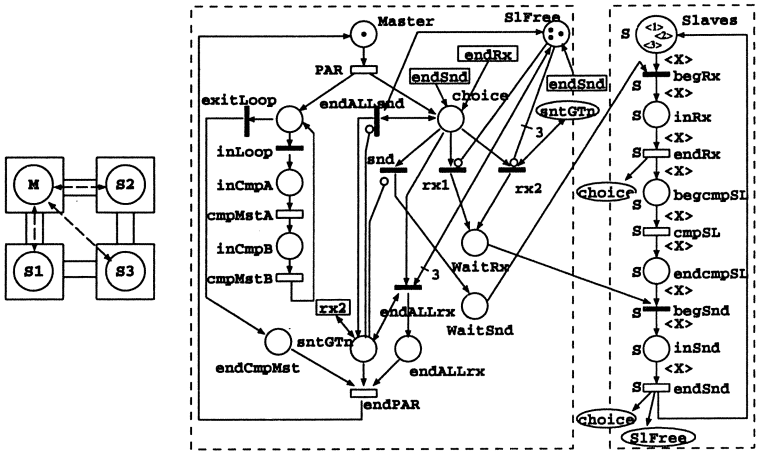


FIGURE 6. The SWN model of the Master-Slave Program.

The simplicity of this example is such that an intuitive argument could be found that would easily allow to compute mean queue length and throughput bounds for this model. In the next section a more complex example is presented in which the presence of synchronization among the system components makes it difficult to find any intuitive argument for bounds computation.

5.2 A simple concurrent program

In this section we present a more complex system consisting of a parallel program organized according to a master-slave computation paradigm. The program is mapped on a parallel architecture: we assume there is a processor for each process, that the processors are homogeneous, and that due to the type of the interconnection network the communication time between the master and the slaves is not homogeneous. This is the case for the example of a master-slave program mapped onto a mesh architecture depicted in Figure 6. The processors are represented by squares, while the processes are represented by circles. In the figure both the physical channels (connecting the processors) and the logical channels (connecting the processes) are depicted.

The master divides the problem to be solved into several subproblems, and as long as there are free slaves it delivers the tasks to be executed to them. When no more free slaves are available, the master waits for some slave to complete its current task. The master also performs some computation in parallel with communicating, to process the results received to prepare new data to be distributed. A SWN model of the master-slaves system is depicted in Figure 6. The model of the program resembles a flow-chart so that its semantics is rather intuitive. The transitions stand for statements in a process code. Places represent pointers to instructions in a given process code.

The model is composed of two subnets, the first (leftmost) one represents the master's behavior while the second (rightmost) one represents the common

behavior of the slaves. The distinction among the three slaves is obtained by using *distinguishable* tokens, i.e., tokens in the slave behavior subnet are labeled with a slave identifier. The behavior model has been simplified as much as possible to make the picture readable. The master is composed of two parts that work in parallel: the computation part, consisting of a certain number of iterations of two procedures (transitions *inLoop*, *cmpMstA*, *cmpMstB*), and the synchronous communication part consisting of a certain number of iterations of send/receive operations (transitions *snd*, *rx1*, *rx2* plus the transitions representing actual communication, shared with the slaves net: *begRx* , *endRx*, *begSnd*, *endSnd*). The slaves behavior can be described as repeated iteration of three operations: receive (transitions *begRx* and *endRx*), compute (transition *cmpSL*), send (transitions *begSnd* and *endSnd*). There is only one color class needed: the slaves class S. It has cardinality three and is divided into two static subclasses $S1$ of cardinality 2 and $S2$ of cardinality 1. Subclass $S1$ represents the slaves that are closer to the master while subclass $S2$ represents the farther slave. As a consequence the transitions representing communication between the master and the slaves have a rate that depends on the slave identity.

The detailed state representation for this model is a list of program counter values, one for each process in the program. This SWN model is quasi-lumpable with respect to the color class of slaves: indeed the qualitative behavior of slaves is identical but the quantitative behavior is not because one slave is farther from the master. We thus aggregate all the states that are equal up to a permutation of slaves identities. It is immediate to verify that the static subclasses $S1$ and $S2$ could be merged. Concerning the symbolic firing instances to be considered for the computation of the lower bound aggregate matrix, we have only to take into account transitions *endRx* and *endSnd*. From the structure of the net it is possible to know in advance that only instances of kind $[endRx, Z_S^j]$ and $[endSnd, Z_S^j]$ with $|Z_S^j| = 1$ are possible so that only two cases have to be considered: $Z_S^j \in S1$ or $Z_S^j \in S2$ so that $min\_rate = com\_rate_1$ and $max\_rate = com\_rate_2$ (where com_rate_i is the inverse of the mean time required for a communication between processing nodes at distance i).

The performance measures we have considered are three: (i) throughput of the system, (THRUPUT); (ii) mean number of slaves waiting to receive a task from the master (MNSWRx); (iii) mean number of slaves waiting to send the result of their computation to the master (MNSWSnd).

We have done three experiments varying the values of rates associated with the communication between master and slaves (transitions *endRx* and *endSnd*). We have fixed the values of $\theta(endRx, S2)$ and $\theta(endSnd, S2)$ to 0.5 and 3.5 respectively, while the rates for $(endRx, S1)$ and $(endSnd, S1)$ that have been used in the three experiments are the following:
$\theta(endRx, S1)$ = (1) 0.51, (2) 0.525, (3) 0.55; $\theta(endSnd, S1)$ = (1) 3.57, (2) 3.675, (3) 3.85.

	Perf. Ind.	Lower bnd	Upper bnd	Spread
1	THRUPUT	0.012396	0.014595	18%
	MNSWSnd	0.079165	0.137257	73%
	MNSWRx	2.547911	2.726692	7%
2	THRUPUT	0.011936	0.016718	40%
	MNSWSnd	0.056813	0.180464	218%
	MNSWRx	2.436226	2.802309	15%
3	THRUPUT	0.011333	0.019553	73%
	MNSWSnd	0.039295	0.257986	557%
	MNSWRx	2.249367	2.861513	27%

The results shown in the table above are obtained from the direct method. We have also applied the bounds improvement method: it resulted in tightened bounds in the first experiment but it did not give any substantial improvement in experiments 2 and 3.

Clearly there is a strict correlation between the difference $\mathcal{Q}^+ - \mathcal{Q}^-$ and the spread in the bounds. It might be worthwhile to assess a priori the sensitivity of the system on variations of the transition probabilities that are positive in $\mathcal{Q}^+ - \mathcal{Q}^-$.

In this example we have also observed that significant bounds refinement could be obtained by the consideration of the high level model (see [4]). Of course, the kind of property to be proved about the high level model in order to obtain improved estimates of $\mathcal{Q}^-$ and/or $\mathcal{Q}^+$ and the technique used to prove it depend heavily on the kind of high level formalism adopted and is difficult to achieve automatically.

6 Conclusions

In this paper an extension to the SRG-based performance analysis technique has been presented for a class of SWN models called *quasi-lumpable* SWNs.

The idea behind this method came from the observation that a system may contain objects that behave homogeneously from a qualitative point of view but the symmetry disappears when quantitative aspects are taken into account. Using the bounds computation method presented in [4] it is possible to take advantage of the stronger aggregation achievable when only the qualitative behavior is taken into account. In this case bounds on the system performance indices are computed instead of exact results; this precision loss can be accepted whether the state space size reduction transforms a model whose state space is too huge to be analyzed into an analyzable model. Observe also that the fact that bounds are obtained instead of approximate results of uncertain precision makes the method more robust than approximation methods especially those that do not give error bounds.

We have shown two application examples, belonging to the class of SWN models just described. Future developments of this work will be in the direction of (1) devising new techniques to further improve the bounds, possibly by using some information from the high level model (e.g. from the SWN

model structural analysis), (2) studying the sensitivity of performance measure bounds spread as a function of the difference $\mathcal{Q}^+ - \mathcal{Q}^-$, and (3) finding new model classes to which the method could apply, as for example those representing systems whose arrival/departure rates are state dependent and are a smooth function of the system population.

References

1. G. Chiola, C. Dutheillet, G. Franceschinis, and S. Haddad. Stochastic well-formed coloured nets for symmetric modeling applications. *IEEE Transactions on Computers*, 42:1343–1360, 1992.
2. P.J. Courtois and P. Semal. Computable bounds on conditional steady-state probabilities in large Markov chains and queueing models. *IEEE Journal on Selected Areas in Communications*, SAC-4(6):926–937, 1986.
3. C. Dutheillet and S. Haddad. Regular stochastic Petri nets. In *Proc. 10th Intern. Conf. Application and Theory of Petri Nets*, Bonn, Germany, June 1989.
4. G. Franceschinis and R. Muntz. Bounds for quasi-lumpable Markov chains. In *Proc. of Performance 93*, Rome, Italy, September 1993.
5. S. Haddad. *Une Categorie Regulier de Reseau de Petri de Haut Niveau: Definition, Proprietes et Reductions.* PhD thesis, Lab. MASI, Université P. et M. Curie (Paris 6), Paris, France, Oct 1987. These de Doctorat, RR87/197 (in French).
6. P. Huber, A.M. Jensen, L.O. Jepsen, and K. Jensen. Towards reachability trees for high-level Petri nets. In G. Rozenberg, editor, *Advances on Petri Nets '84*, volume 188 of *LNCS*, pages 215–233. Springer Verlag, 1984.
7. K. Jensen. Coloured Petri nets and the invariant method. *Theoretical Computer Science*, 14:317–336, 1981.
8. J.G. Kemeny and J.L. Snell. *Finite Markov Chains.* Van Nostrand, Princeton, NJ, 1960.
9. Chuang Lin and Dan C. Marinescu. On stochastic high level Petri nets. In *Proc. Int. Workshop on Petri Nets and Performance Models*, Madison, WI, USA, August 1987. IEEE-CS Press.

Functional and Performance Analysis of Cooperating Sequential Processes[1]

J. Campos J.M. Colom M. Silva E. Teruel
Departamento de Ingeniería Eléctrica e Informática
Centro Politécnico Superior, Universidad de Zaragoza
María de Luna 3, E-50015 Zaragoza, Spain

This paper presents some results concerning the functional and performance analysis of sequential processes connected through buffers using structural analysis techniques, mainly linear algebraic ones. From the functional point of view the following properties are considered: boundedness, deadlock-freeness, liveness and the existence of home states. From the performance point of view the considered properties are marking ergodicity, computation of visit ratios and computation of insensitive throughput bounds.

1 Introduction

The design of distributed systems is usually a complex task, compelling the use of formal methods. A major trend in the modelling of concurrent and distributed systems is the use of a single formalism during the entire design process [10]. Such formalism should provide:

- Basic modelling features like: simple primitives for the modelling of *sequence*, *choice*, and *concurrency*; a powerful communication support for designers; hierarchical and modular modelling methodologies; the possibility of parameterization of models...
- A well founded logical theory providing the definition of functional properties like deadlock-freeness or the absence of (buffer) overflows, and validation algorithms for them.
- A natural representation of time and the possibility of qualitative and quantitative analysis of performance properties.

In [17] it is claimed that the interleaving of functional and performance theories for the analysis of systems produces important benefits for both kinds of analysis. The present work supports such claim.

[1]This work has been partially supported by the projects P IT-6/91 of the Aragonese CONAI (DGA), CICYT TIC-91-0354 and ROB-91-0949 of the Spanish Plan Nacional de Investigación, and Esprit BRA Project 7269 (QMIPS) and W.G. 6067 (CALIBAN).

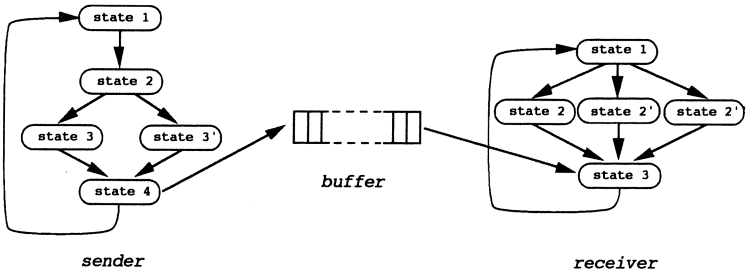


FIGURE 1. Two sequential processes communicating through a buffer.

At present, two modelling paradigms satisfying the above requirements are available in the literature: that based on *programming language constructs*, like CCS [12] or CSP [11], and that based on *graphical constructs*, like Petri nets [3, 16, 14], extended with the corresponding time representations. In this paper we consider the latter and, in particular, we concentrate on systems obtained by the application of a simple modular design principle: several *sequential processes* execute concurrently and *cooperate using asynchronous communication by message passing through a set of buffers.* (The restrictions imposed upon the connectivity of buffers prevent *competition.*) As soon as a sequential process S, the *sender*, needs to communicate with another sequential process R, the *receiver*, S deposits *messages* (tokens) in a buffer, the medium of communication. When R is ready, if there are enough messages in the corresponding buffer, R takes them (see Figure 1). The possible information contained in messages can be disregarded, paying attention to the control flow only. In other words, messages can be considered as *authorizations.* Application domains where this class of systems appears are computer networks, information systems, operating systems, real-time systems, nonsequential programming languages, and discrete part manufacturing systems, among others.

Several works exist concerning functional and performance analysis of systems of sequential processes communicating through buffers modelled with Petri nets. Various aspects of modelling and functional analysis can be found in [15, 18, 19], including definitions of different subclasses, structural analysis of properties, compositionality concepts, etc. A first approach to efficient (with polynomial time complexity on the net size) performance analysis was presented in [6], stressing both functional and performance aspects. Two results were presented in that work: a characterization of ergodicity of the marking process for certain subclasses with exponentially distributed service times of transitions, in terms of two structural properties, namely consistency and some synchronic distance relations; and a polynomial time algorithm to compute the exact throughput of transitions in the steady state.

In this paper we try to bridge qualitative and quantitative aspects of *Deterministic Systems of Sequential Processes*, with the goal of obtaining benefits

in both the validation of functional properties and the evaluation of performance indices of such net systems. The paper is organized as follows: in Section 2 the class of Deterministic Systems of Sequential Processes is defined. Section 3 deals with some aspects concerning the functional analysis, and Section 4 considers performance properties such as ergodicity and the computation of quantitative indices.

2 Deterministic Systems of Sequential Processes

In this section we formally define the class of Deterministic Systems of Sequential Processes as a subclass of Petri net systems, and we explain how time is introduced in the model. Previously, some basic definitions and notations of Petri nets are recalled.

The class that we consider in this paper is an extension of that introduced in [18], although we keep the same name. In that work, sequential processes are modelled with *safe State Machines* while the communication among them is described by their connection through particular places called *buffers*: their buffers are private in the sense that each of them has only one input and only one output State Machine. Our extension allows that several State Machines deposit messages (tokens) in a buffer.

2.1 Basic Definitions and Notations of Petri Nets

Let us recall some definitions and notations about Petri nets (we refer the reader to [3, 16, 14] for more comprehensive presentations).

A *P/T net* is a 4–tuple $\mathcal{N} = (P, T, Pre, Post)$, where P and T are disjoint sets of *places* and *transitions* ($|P| = n$, $|T| = m$), and Pre ($Post$) is the *pre- (post-) incidence function* representing the input (output) arcs: $Pre: P \times T \rightarrow \mathbb{N} = \{0, 1, 2, \ldots\}$ ($Post: P \times T \rightarrow \mathbb{N}$). A Petri net can be seen as a bipartite directed graph in which places and transitions are the two kinds of nodes. Places are usually drawn as circles while transitions are depicted as bars or boxes. *Ordinary* nets are Petri nets whose pre- and post-incidence functions take values in $\{0, 1\}$. The incidence function of a given arc in a non-ordinary net is called *weight* or *multiplicity*. The *pre-* and *post-set* of a transition $t \in T$ are defined respectively as ${}^\bullet t = \{p | Pre(p,t) > 0\}$ and $t^\bullet = \{p | Post(p,t) > 0\}$. The *pre-* and *post-set* of a place $p \in P$ are defined respectively as ${}^\bullet p = \{t | Post(p,t) > 0\}$ and $p^\bullet = \{t | Pre(p,t) > 0\}$. The *incidence matrix* of the net is defined as $C = [Post(p_i, t_j) - Pre(p_i, t_j)]$, $i = 1, \ldots, n$, $j = 1, \ldots, m$. Similarly the *pre- and post-incidence matrices* are defined as $PRE = [Pre(p_i, t_j)]$ and $POST = [Post(p_i, t_j)]$. *Flows* (*semiflows*) are integer (natural) annullers of C. Right and left annullers are called T- and P-(semi)flows respectively. A semiflow is called *minimal* when its support, $\|X\|$, is not a proper superset of the support of any other, and the greatest common divisor of its elements is one. Unless explicitly stated, we shall not consider the trivial flow, i.e. vector 0. Flows are important because they induce certain invariant relations which are useful for reasoning on the behaviour. Actually, several structural properties are defined in terms of the existence of certain

annullers, or similar vectors:

$$\begin{aligned} \mathcal{N} \text{ is consistent (structurally repetitive)} &\Leftrightarrow \\ \exists X \geq \mathbb{1} \text{ such that } C \cdot X = (\geq)\, 0 & \\ \mathcal{N} \text{ is conservative (structurally bounded)} &\Leftrightarrow \\ \exists Y \geq \mathbb{1} \text{ such that } Y \cdot C = (\leq)\, 0 & \end{aligned}$$

A function $M: P \rightarrow \mathbb{N}$ (usually represented in vector form) is called *marking*. A *P/T system*, or *marked Petri net*, $(\mathcal{N}, M_0)$, is a P/T net $\mathcal{N}$ with an *initial marking* M_0. A transition $t \in T$ is *enabled* at marking M iff $\forall p \in P$: $M(p) \geq Pre(p,t)$. A transition t enabled at M can *fire* yielding a new marking M' (*reached* marking) defined by $M'(p) = M(p) - Pre(p,t) + Post(p,t)$ (it is denoted by $M \xrightarrow{t} M'$). A sequence of transitions $\sigma = t_1 t_2 \ldots t_n$ is a *firing sequence* in $(\mathcal{N}, M_0)$ iff there exists a sequence of markings such that $M_0 \xrightarrow{t_1} M_1 \xrightarrow{t_2} M_2 \ldots \xrightarrow{t_n} M_n$. In this case, marking M_n is said to be *reachable* from M_0 by firing σ, and this is denoted by $M_0 \xrightarrow{\sigma} M_n$. The function $\vec{\sigma}: T \rightarrow \mathbb{N}$ is the *firing count vector* of the firable sequence σ, i.e. $\vec{\sigma}[t]$ represents the number of occurrences of $t \in T$ in σ. If $M_0 \xrightarrow{\sigma} M$, then we can write in vector form $M = M_0 + C \cdot \vec{\sigma}$, which is referred to as the *linear state equation* of the net. A marking M is said to be *potentially reachable* iff $\exists \vec{\sigma} \geq 0$ such that $M = M_0 + C \cdot \vec{\sigma} \geq 0$. The *reachability set* $R(\mathcal{N}, M_0)$ is the set of all markings reachable from the initial marking. Denoting by $PR(\mathcal{N}, M_0)$ the set of all potentially reachable markings we have the following relation: $R(\mathcal{N}, M_0) \subseteq PR(\mathcal{N}, M_0)$.

A place $p \in P$ is said to be *k–bounded* iff $\forall M \in R(\mathcal{N}, M_0)$, $M(p) \leq k$. A P/T system is said to be (marking) k–bounded iff every place is k–bounded, and bounded iff there exists some k for which it is k–bounded. A P/T system is *live* when every transition can ultimately occur from every reachable marking, and it is *deadlock-free* when at least one transition is enabled at every reachable marking. M is a *home state* in $(\mathcal{N}, M_0)$ iff it is reachable from every reachable marking, and $(\mathcal{N}, M_0)$ is *reversible* iff M_0 is a home state. The *home space* is the set of home states. Boundedness is necessary whenever the system is to be implemented, while liveness is often required, specially in reactive systems. They are so important that the name *well-behaved* has been coined for live and bounded systems. A net $\mathcal{N}$ is *structurally bounded* when $(\mathcal{N}, M_0)$ is bounded for *every* M_0, and it is *structurally live* when *there exists* an M_0 such that $(\mathcal{N}, M_0)$ is live. Consequently, if a net $\mathcal{N}$ is structurally bounded and structurally live there exists some marking M_0 such that $(\mathcal{N}, M_0)$ is well-behaved. In such case, non well-behavedness is exclusively imputable to the marking, and we say that the net is *well-formed*. A well-known polynomial necessary condition for well-formedness, based solely on purely structural properties (i.e. properties that can be defined without any reference to the behaviour) is structural boundedness and structural repetitiveness. For convenience sake, a structurally bounded and structurally repetitive net will be called *well-structured*. Some well-known relations between these concepts are summarized as follows [3, 16, 14, 4]: Let

$(\mathcal{N}, M_0)$ be a connected P/T system. If $\mathcal{N}$ is well-formed, then it is well-structured, which is equivalent to consistent and conservative. If $\mathcal{N}$ is well-structured, then it is strongly connected. If $(\mathcal{N}, M_0)$ is well-behaved, then $\mathcal{N}$ is strongly connected and consistent.

2.2 Deterministic Systems of Sequential Processes, and Other Subclasses

State Machines are ordinary Petri nets such that every transition has only one input and only one output place ($\forall t \in T$: $|{}^\bullet t| = |t^\bullet| = 1$). State Machines allow the modelling of sequences, decisions (or conflicts), and re-entrancy (when they are marked with more than one token) but not synchronization. Some well-known results from the structure theory of State Machines are the following [3, 16, 14]:

- The rank of their incidence matrix equals their number of places minus one
- They are conservative (thus, structurally bounded)
- A State Machine is structurally live iff it is strongly connected, which is equivalent to being consistent
- A marked State Machine is live iff it is strongly connected and it contains at least one token
- If a marked State Machine is live, then it is k–bounded iff it contains k tokens.

(Regarding the timing, it is assumed that every cycle in a State Machine contains at least one timed transition.) Topologically speaking, strongly connected State Machines are the Petri net counterpart of classical closed monoclass queueing networks. In *closed* networks, no customer leaves the system or arrives from the outside, hence the population is preserved. The corresponding property in Petri nets terminology is *conservativeness*, which leads to global token conservation laws for any initial marking.

Deterministic Systems of Sequential Processes are used for the modelling and analysis of distributed systems composed by sequential processes communicating through output-private buffers. Each sequential process is modelled by a safe (1–bounded) State Machine. The communication among them is described by *buffers* (places) which contain *products/messages* (tokens), which are produced by some processes and consumed by others. Each buffer is *output-private* in the sense that it is an input place of only one State Machine (see Figure 2, where grey places are buffers).

DEFINITION 1 *A marked Petri net* $(\mathcal{N}, M_0) = (P_1 \cup \ldots \cup P_q \cup B, T_1 \cup \ldots \cup T_q, Pre, Post, M_0)$ *is a* Deterministic System of Sequential Processes (DSSP) *iff:*

1. $\forall i, j \in \{1, \ldots, q\}, i \neq j$: $P_i \cap P_j = \emptyset$, $T_i \cap T_j = \emptyset$, $P_i \cap B = \emptyset$

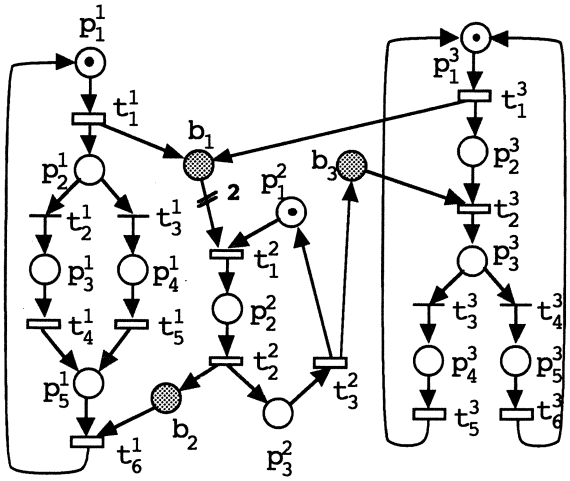


FIGURE 2. A Deterministic System of Sequential Processes.

2. $\forall i \in \{1,\ldots,q\}$: $(\mathcal{SM}_i, M_{0i}) = (P_i, T_i, Pre_i, Post_i, M_{0i})$ *is a strongly connected and 1-bounded State Machine (where* Pre_i,$Post_i$, *and* M_{0i} *are the restrictions of* Pre, $Post$, *and* M_0 *to* P_i *and* T_i*)*

3. *The set* B *of buffers is such that* $\forall b \in B$*:*

   (a) $|{}^\bullet b| \geq 1$ *and* $|b^\bullet| \geq 1$

   (b) $\exists i \in \{1,\ldots,q\}$, *such that* $b^\bullet \subset T_i$

   (c) $\forall p \in P_1 \cup \ldots \cup P_q$: $t, t' \in p^\bullet \Rightarrow Pre(b,t) = Pre(b,t')$.

The first two items of the previous definition state that a DSSP is composed by a set of State Machines ($\mathcal{SM}_i, i = 1,\ldots,q$) and a set of buffers ($B$). By item *3.a*, buffers are neither source nor sink places. The output-private condition is expressed by condition *3.b*. Requirement *3.c* justifies the word "deterministic" in the name of the class: the marking of buffers does not disturb the decisions taken by a State Machine, i.e. choices in the State Machines are free. This definition generalizes the class of DSSP's defined by Souissi and Beldiceanu [18], where buffers are required to have not only a single output State Machine (output-private) but also a single input State Machine (input-private). From a queueing network perspective, DSSP's are a mild generalization of *Fork-Join Queuing Networks with Blocking* where servers are complex (safe State Machines with a rich connectivity to buffers).

From the definition of a DSSP, it follows that the incidence matrix has the following structure:

$$C = \begin{array}{c} \begin{array}{cccc} T_1 & T_2 & \cdots & T_q \end{array} \\ \left(\begin{array}{c|c|c|c} C_1 & 0 & \cdots & 0 \\ \hline 0 & C_2 & \cdots & 0 \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline 0 & 0 & \cdots & C_q \\ \hline B_1 & B_2 & \cdots & B_q \end{array}\right) \end{array} \begin{array}{c} P_1 \\ P_2 \\ \vdots \\ P_q \\ B \end{array} \tag{1}$$

where C_i represents the incidence matrix of $\mathcal{SM}_i = (P_i, T_i, Pre_i, Post_i)$, while B_j represents the connections of State Machine $\mathcal{SM}_j$ to the buffers.

From the particular structure of C, the following can be written:

$$\text{rank}(C) \leq \sum_{i=1}^{q} \text{rank}(C_i) + \text{rank}(C_B) = \sum_{i=1}^{q} |P_i| - q + \text{rank}(C_B) \tag{2}$$

where $C_B = (B_1|B_2|\cdots|B_q)$ is the submatrix of C formed by the rows corresponding to the buffers.

Another interesting subclass of P/T nets are *Equal Conflict* nets [22]:

DEFINITION 2 *Let $\mathcal{N}$ be a P/T net. Two transitions, $t, t' \in T$, are in* Equal Conflict *relation iff $t = t'$ or ${}^\bullet t \cap {}^\bullet t' \neq \emptyset \Rightarrow \forall p \in P: Pre(p,t) = Pre(p,t')$. This is an equivalence relation on the set of transitions of a net, and every equivalence class is called an Equal Conflict (set). The set of all Equal Conflict sets is denoted by $\mathcal{E}$.*

$\mathcal{N}$ is an Equal Conflict (EC) *net iff $\forall t, t' \in T: {}^\bullet t \cap {}^\bullet t' \neq \emptyset \Rightarrow \forall p \in P: Pre(p,t) = Pre(p,t')$.*

EC nets are such that every choice is free, so they generalize the ordinary subclass of Free Choice nets. Many nice results from the Free Choice theory have been recently extended to EC net systems:

- [22] The potential reachability graph of a live EC system is *directed.* Thus, live EC systems do not have *killing spurious solutions* (spurious solutions that do not enable any transition), and live and bounded EC systems have home states. Since a bounded strongly connected EC system is live iff it is deadlock-free, liveness can be determined by checking absence of solutions to certain systems of linear inequalities in the integer domain [20]

- [23] $\mathcal{N}$ is well-formed iff it is consistent, conservative, and the rank of its incidence matrix fulfills some simple condition based on the topology. Therefore, the possibility of marking boundedly and lively a given EC net can be determined in polynomial time. Moreover, if $\mathcal{N}$ is well-formed, it can be decomposed in a meaningful way, similarly to the decomposition of a Free Choice net into State Machines and/or Marked Graphs. Liveness of the whole system can be compositionally characterized in terms of the liveness of the analogues to State Machine components.

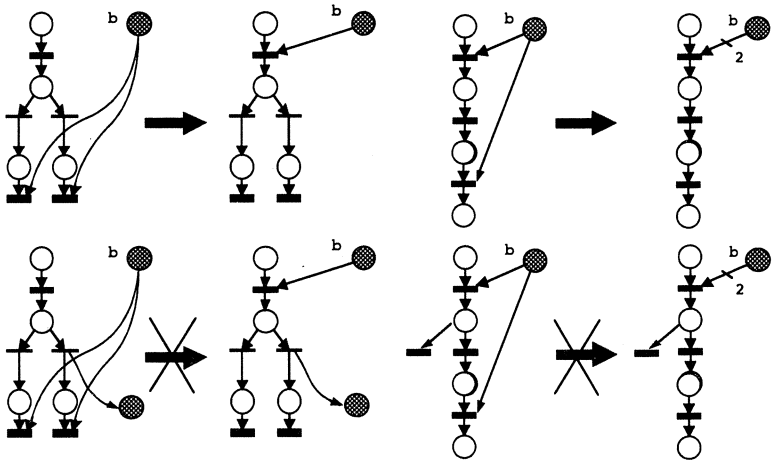


FIGURE 3. Example of transformations (preserving boundedness, liveness, the existence of home states, etc.) to convert a DSSP into a DSSP/EC. Some DSSP nets cannot be transformed, though.

Observe that, in a DSSP, all the private conflicts of the State Machines are Equal Conflicts by the "deterministic" assumption. All the remaining transitions are elementary Equal Conflict sets. In the sequel, for ease of presentation, Equal Conflict sets will be supposed to have at most two transitions (otherwise serialize choices into binary ones).

Neither DSSP's are a subclass of EC nets nor the converse. Nevertheless, if it happens that all buffers of a DSSP have exactly one output transition ($\forall b \in B: |b^\bullet| = 1$), that is, if they are *strictly* output-private, then obviously they are EC, and they are called *Equal Conflict DSSP (DSSP/EC)* (the net in Figure 2 is indeed EC). Naturally, DSSP/EC's inherit all the nice properties of EC nets and systems. In fact, by means of several transformations preserving, among other properties, boundedness, liveness and the existence of home states (see Figure 3 top for two examples), the results that are valid for the DSSP/EC subclass can be extended to many non EC nets. Observe, though, that *not* all DSSP's can be transformed into EC (see Figure 3 bottom).

2.3 Time Representation

One of the advantages of Petri net models for the design and analysis of concurrent and distributed systems is that they can be naturally extended by time attributes in order to achieve performance evaluation. We consider net systems with timed transitions. Marking and time independent Coxian random variables associated to the firing of transitions define their *service time*. The mean values of these variables are denoted s_i for each transition t_i of the net.

For the modelling of conflicts we use *immediate transitions* with the addi-

tion of (marking and time independent) *routing rates* [1]. In other words, for the subset of immediate transitions $\{t_1, \ldots, t_k\} \subset T$ being in conflict at each reachable marking, we assume that the constants $r_1, \ldots, r_k \in \mathbb{Q}^+$ are explicitly defined in the system interpretation in such a way that when $t_1, \ldots, t_k$ are simultaneously enabled, transition t_i, $i = 1, \ldots, k$, fires with relative rate $r_i/(\sum_{j=1}^{k} r_j)$. Consequently, routing is completely decoupled from duration of activities. The only restriction that this decoupling imposes upon the system is that *preemption* cannot be modelled with two timed transitions (in conflict) competing for the tokens. (In other words, a *race policy* cannot be modelled. Our constraint is equivalent to the use of a *preselection policy* for the resolution of conflicts among timed transitions.)

Assuming the above described time interpretation, the timed model has almost surely the *fair progress* property, that is, no transition can be permanently enabled without firing. Additionally, it has the *local fairness* property, that is, all output transitions of a shared place simultaneously enabled at infinitely many markings will fire infinitely often. (In other words, all possible outcomes of any conflict have a non-null probability of firing.)

The *visit ratio* of transition t_i with respect to t_j, $v_i^{(j)}$, is the average number of times t_i is visited (fired) for each visit to (firing of) the reference transition t_j. The computation of visit ratios is interesting for the performance analysis of formal models. For example, it is well-known that the steady-state probability of a state in a product-form queueing network with single-server semantics [9] depends on the *average service demands* of customers from station i, defined as:

$$D_i^{(j)} \stackrel{def}{=} v_i^{(j)} \cdot s_i \qquad i = 1, \ldots, m \tag{3}$$

The computation of average service demands is also very important in the performance analysis of stochastic Petri net models. In Section 4, applying the theory presented in [8], we use these values to compute upper and lower bounds for the *throughput* of transitions, i.e. the average number of service completions (firings) per time unit, in a well-behaved DSSP.

3 Functional Analysis of DSSP's

Associated to the net $\mathcal{N}$ of a DSSP, two nets, the Regulated Net and the Control Net, can be defined. In the first one, a particular conflict resolution policy respecting the relative occurrence of transitions in (private) conflicts, is implemented by means of some "arbiter" subnets, that reduce the non-determinism. The second one, derived from the first, tries to capture the essentials of the intercommunication schema, somehow abstracting from the details of the sequential processes. One of the benefits of using restricted subclasses of nets is the availability of results that facilitate the analysis. In the case of DSSP, we have a special necessary condition for well-formedness, that we conjecture to be also sufficient (actually, it is proven to be for DSSP/EC), and a simple algebraic sufficient condition for liveness, based on the equivalence of liveness and deadlock-freeness (it is also proven to be necessary for DSSP/EC).

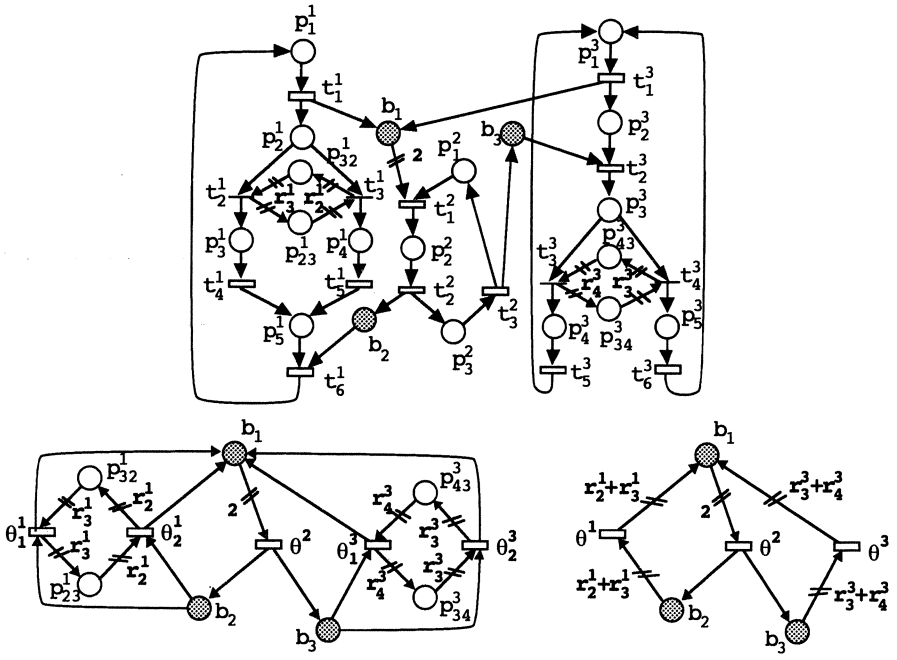


FIGURE 4. Regulated and Control Nets.

3.1 The Regulated Net, $\mathcal{N}_R$, and the Control Net, $\mathcal{N}_C$

The *regulated net* associated to $\mathcal{N}$ for some given routing rates, $\mathcal{N}_R$, is obtained adding to $\mathcal{N}$ an appropriate *Regulation Circuit* [4] per (binary) Equal Conflict set (i.e. a circuit around the two conflicting transitions, the weights of arcs being such that the transitions should be fired at long run according to the routing rates; see Figure 4 top). It is easy to see that, if $(\mathcal{N}, M_0)$ is live, it is always possible to mark appropriately the Regulation Circuits, so that $(\mathcal{N}_R, M_0^R)$ is also live ($M_0 = M_0^R \downarrow_P$, i.e. the *projection* of M_0^R on P).

Let us now define $\mathcal{N}_C$, the *parametrically weighted control net associated to* $\mathcal{N}$ (the *control net* in the sequel). Let $\mathcal{N}_R^*$ be the net in which the routing rates are leaved in parametric form (i.e. the rates r_i are not given a numerical value). Apply (partially) the basic algorithm to compute all minimal T-semiflows of a net [13, 7] with the aim of eliminating only certain rows (places):

Phase 1. Eliminate places belonging to the State Machines (Figure 4 bottom, left)

Phase 2. Eliminate places belonging to the Regulation Circuits (Figure 4 bottom, right)

The outcome of Phase 1 is a matrix, C_1, that in net terms can be interpreted as follows:

- Each transition represents a minimal T-semiflow of a State Machine
- Places are the buffers and those derived from the Regulation Circuits.

The outcome of Phase 2 is another matrix, C_2. The submatrix of C_2 obtained removing the null rows, i.e. considering only the rows corresponding to buffers: $C_C = C_2 \downarrow_B$, is of course the incidence matrix of a P/T net, and this will be precisely the *control net* $\mathcal{N}_C$:

- Each transition represents a State Machine of the DSSP
- All places are buffers
- The weights of arcs are parametrized by the routing rates.

PROPOSITION 3 *Let $\mathcal{N}$ be a well-structured DSSP with q State Machines:*

- *$\mathcal{N}_C$ is a strongly connected and conservative structurally persistent net (or Choice-free, see [21]) with parametric weights*
- $\text{rank}(C_C) = q - 1$ *or* $\text{rank}(C_C) = q$.

It is interesting to observe that all the T-semiflow structure of $\mathcal{N}$ (derived from the Equal Conflicts) is represented in the control net thanks to the parametric weighting, as can be checked working out the example of Figures 2 and 4.

From the above, an important lower bound for $\text{rank}(C)$ is obtained:

PROPOSITION 4 *Let $\mathcal{N}$ be a well-structured DSSP, with incidence matrix C, and with $|\mathcal{E}|$ Equal Conflict sets:*

$$|\mathcal{E}| - q + \text{rank}(C_B) \geq \text{rank}(C) \geq |\mathcal{E}| - 1 \tag{4}$$

Proof. The left inequality is simply a rewriting of inequality (2), taking into account that, for strongly connected State Machines, there is exactly one place per Equal Conflict set, so $\sum_{i=1}^{q} |P_i| = |\mathcal{E}|$.
For the right inequality the following matrix should be considered:

$$(\; C \mid C_2 \;) \stackrel{def}{=} \left(\begin{array}{c|c} \{C_i\} & 0 \\ \hline C_B & C_C \end{array} \right)$$

Since C_2 is obtained through linear combinations of columns of C, $\text{rank}(C|C_2) = \text{rank}(C)$. Therefore:

$$\text{rank}(C) \geq \sum_i |P_i| - q + \text{rank}(C_B) \geq |\mathcal{E}| - q + q - 1 = |\mathcal{E}| - 1 \tag{5}$$

◇

3.2 Well-Formedness and Liveness

Using the bound obtained in Proposition 4, and also the particular structure of DSSP's we obtain the following result regarding liveness of a DSSP:

THEOREM 5 *Let* $(\mathcal{N}, M_0)$ *be a DSSP.*

- *If* $\mathcal{N}$ *is well-formed, then* $\mathcal{N}$ *is well-structured and* $\text{rank}(C) = |\mathcal{E}| - 1$
- *If* $(\mathcal{N}, M_0)$ *is strongly connected and bounded, it is live iff it is deadlock-free.*

Proof. For Part 1, according to [4], $\text{rank}(C) \leq |\mathcal{E}| - 1$ for any well-formed P/T net. But $\text{rank}(C) \geq |\mathcal{E}| - 1$ for a DSSP.
For Part 2, if t_j^k is a non-live transition of SM_k, none of the transitions of SM_k would be live. Since the input buffers to SM_k are bounded by assumption, the transitions of all their input State Machines should be non-live. By finiteness and strong connectedness of $\mathcal{N}$, non-liveness is propagated to all transitions, hence the system is deadlocked. ◇

If $\mathcal{N}$ is not well-structured, then it cannot be well-formed. If $\mathcal{N}$ is well-structured but $\text{rank}(C) \geq |\mathcal{E}|$, then it is not structurally live (part 1 of Theorem 5) and for every initial marking the system would deadlock (part 2 of Theorem 5). In performance terms, in both cases, assuming boundedness, the system would present null throughput for any initial configuration of resources/customers due to a problem that is rooted on the net structure. The problem can be detected in polynomial time (both well-structuredness and the rank are analysed in polynomial time), so this test should be applied prior to any other — more complex — analysis. For DSSP/EC, the condition of part 1 of Theorem 5 is also sufficient [23], result that can be extended to many non EC nets by the corresponding net transformations (see Figure 3). We conjecture that the result is valid for the complete DSSP class.

In case $\mathcal{N}$ is well-formed, the problem is determining whether the initial marking makes the system live or not. To achieve this, part 2 of Theorem 5 can be used, so only deadlock-freeness needs to be proven, instead of liveness. (We want to mention here that being well-formed is not proven sufficient, in general, to guarantee existence of an initial marking making the system live and bounded *and DSSP*, due to the definition imposing safeness of the State Machines. Nevertheless, in the case of DSSP/EC, and thanks to the compositional characterization of liveness [23], if a DSSP/EC is well-formed then there exists a "DSSP marking" such that the system is well-behaved, because one token is sufficient for the P-components corresponding to State Machines to be live.) In [20] a general sufficient condition for deadlock-freeness in terms of the absence of integer solutions to a set of systems of linear inequalities is presented. The basic idea is to ask for absence of potentially reachable markings not enabling any transition of the net. In the particular case of DSSP's, among other subclasses, such algebraic condition can be expressed as a single system of inequalities (use the rules presented in [20] plus the particular transformation

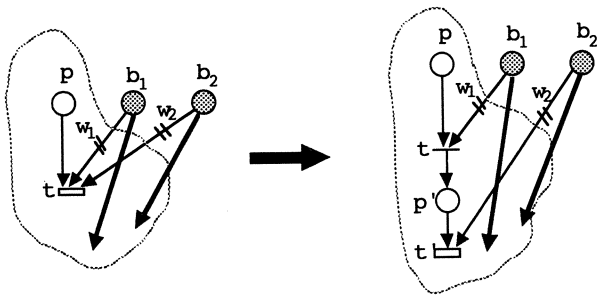


FIGURE 5. A transformation preserving deadlock-freeness. (Place p must *not* be a choice place.)

shown in Figure 5, that preserves deadlock-freeness — actually it preserves the language modulo a projection — thanks to the output-private and the State Machines' safeness hypothesis).

For instance, absence of (potential) deadlocks in the system of Figure 2 is characterized by the absence of solutions to the following system of inequalities. The first two sets define the set of potentially reachable markings (i.e. they are the state equation). The following three sets express disabledness of "private" transitions of the three State Machines (i.e. transitions whose only input is a State Machine place), and the last three express disabledness of the transitions having some buffer as input ($SB(b)$ denotes the *structural (marking) bound* of buffer b, that is defined as $\max\{M(b) \mid M \in PR(\mathcal{N}, M_0)\}$; the structural bound of State Machine places is obviously one).

$M = M_0 + C \cdot \vec{\sigma}$
$M \geq 0; \vec{\sigma} \geq 0$
$M(p_1^1) = M(p_2^1) = M(p_3^1) = M(p_4^1) = 0$
$M(p_2^2) = M(p_3^2) = 0$
$M(p_1^3) = M(p_3^3) = M(p_4^3) = M(p_5^3) = 0$
$SB(b_2) \cdot M(p_5^1) + M(b_2) \leq SB(b_2)$
$SB(b_1) \cdot M(p_1^2) + M(b_1) \leq SB(b_1) + 1$
$SB(b_3) \cdot M(p_1^2) + M(b_1) \leq SB(b_3)$

This general sufficient condition for deadlock-freeness is also necessary in the case of EC systems [22], and we conjecture that it is so also for DSSP's. (Safeness of the State Machines is necessary here. There are examples of DSSP nets with a 2–bounded State Machine having killing spurious solutions.)

4 PERFORMANCE ANALYSIS OF DSSP'S

4.1 Home States and Ergodicity

It is well-known that under (possibly marking-dependent) exponentially distributed random variables associated to the firing of transitions and Bernoulli trials for the successive resolutions of each conflict, the underlying *Continuous*

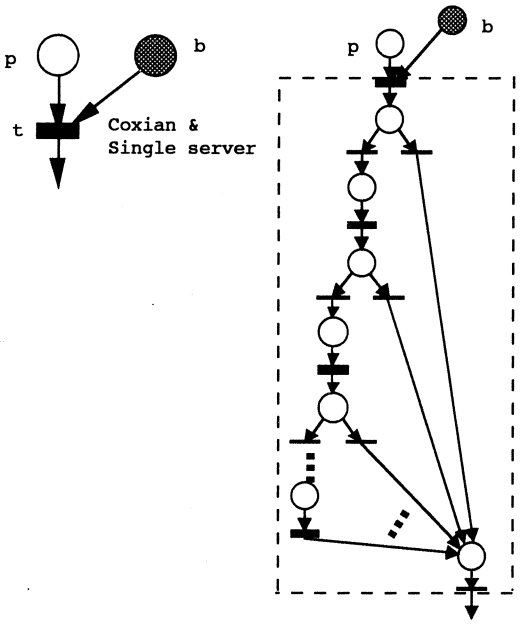


FIGURE 6. A refinement into exponential stages of a Coxian single server timing of t.

Time Markov Chain (CTMC) is isomorphous to the reachability graph of the untimed net model [2]. Thus, the existence of a home space leads to ergodicity of the marking process for bounded net systems with exponential firing times.

PROPERTY 6 *Let $(\mathcal{N}, M_0)$ be a live and bounded DSSP/EC with Coxian random variables associated to the firing of transitions and Bernoulli trials for the successive resolutions of each conflict. The underlying CTMC is ergodic.*

Proof. Live and bounded Equal Conflict net systems have home states [22]. By 1-boundedness of the State Machines of a DSSP (i.e. they work under single server semantics) two transitions of the same State Machine cannot be fired simultaneously. Therefore, the refinement of a Coxian-timed single-server transition using exponentially timed stages can be applied (Figure 6). Since the (refined) exponentially timed net system is a DSSP/EC, its underlying CTMC is ergodic. ◇

Although the property is stated for DSSP/EC, again the result can be extended to many non EC nets by the corresponding net transformations preserving the existence of home states (see Figure 3). We conjecture that DSSP have also home states, that is, that the ergodicity result holds for the entire DSSP class of models.

4.2 Computation of Visit Ratios

The computation of the average service demands of tokens from transitions, Equation (3), is useful for the performance analysis of timed systems. Assuming that the average service times of transitions, s_i, are known, then it is necessary to compute the vector of visit ratios to transitions, $\vec{v}^{(j)}$.

If $\mathcal{N}$ is well-structured, the visit ratio vector normalized for t_j, $\vec{v}^{(j)}$, should be a T-semiflow of $\mathcal{N}$. Otherwise stated (observe that $v_i^{(j)} \geq 0$ by definition):

$$C \cdot \vec{v}^{(j)} = 0 \tag{6}$$

The conflicts in the State Machines of a DSSP are free, because the buffers do not condition the conflict resolution, by the "deterministic" assumption. Structurally speaking, these conflicts correspond to Equal Conflicts at the net level. Let t_a and t_b be in Equal Conflict relation. The corresponding visit ratios should verify the following equation:

$$r_b \cdot v_a^{(j)} - r_a \cdot v_b^{(j)} = 0 \tag{7}$$

An equation like (7) holds for every (binary) Equal Conflict. Rewritten in vector form: $r_{ab} \cdot \vec{v}^{(j)} = 0$, which for the set of all Equal Conflicts leads to:

$$R \cdot \vec{v}^{(j)} = 0 \tag{8}$$

where R is a matrix with $m - |\mathcal{E}|$ (number of binary Equal Conflicts) rows and m columns. In other words, $m - |\mathcal{E}|$ is the number of independent linear relations fixed by the routing rates at (binary) Equal Conflicts, so $\text{rank}(R) = m - |\mathcal{E}|$.

THEOREM 7 *Let $\mathcal{N}$ be a well-formed DSSP net. The system of equations:*

$$\begin{pmatrix} C \\ R \end{pmatrix} \cdot \vec{v}^{(j)} = 0,\ v_j^{(j)} = 1 \tag{9}$$

has only one solution (i.e. the vector of visit ratios depends neither on the marking, provided it allows infinite behaviours, nor on the service times).

Proof. The above statement is equivalent to saying that $\mathcal{N}_R$ has a unique consistent component (in fact it is consistent) under locally fair R and $\text{rank} \begin{pmatrix} C \\ R \end{pmatrix} = m - 1$. If $\mathcal{N}$ is well-formed, the addition of a Regulation Circuit to a (binary) Equal Conflict increases by one the rank of the incidence matrix [4]. To produce $\mathcal{N}_R$ from $\mathcal{N}$, $m - |\mathcal{E}|$ regulation circuits must be added, so:

$$\text{rank} \begin{pmatrix} C \\ R \end{pmatrix} = \text{rank}\,(C) + m - |\mathcal{E}| = m - 1 \tag{10}$$

The addition of regulation circuits preserves well-formedness, thus consistency. ◇

Consistency of $\mathcal{N}_R$ means that in any infinitely long run, all transitions appear infinitely often. In other words, this fact can be enlightened as follows:

Let $\mathcal{N}$ be a well-formed DSSP net. Any locally fair conflict resolution policy for an M_0 such that $(\mathcal{N}, M_0)$ is live leads to global fairness (impartiality). If the system of inequalities (4) has no solution for a given conservative DSSP net and for some given routing rates defined by R_0, $\mathcal{N}_{R_0}$ should not contain a consistent component and there is no possibility of infinite behaviours. Thus $\mathcal{N}$ is not structurally live and a total deadlock can be reached sooner or later. Moreover, $\mathcal{N}$ being conservative, $(\mathcal{N}, M_0)$ should be bounded, then if conflicts are solved under time and marking independent discrete probability distributions and there is no null probability of firing an enabled transition (i.e. local fairness is assumed), the stochastic net system will *inevitably* reach a deadlock.

4.3 Performance Bounds

This section is devoted to present some *insensitive* (i.e. holding for any probability distribution function for the firing times) performance bounds. Basically, throughput upper bounds are computed by finding the slowest isolated subnet among those generated by P-semiflows of the net, and are presented in the next theorem.

THEOREM 8 [8]

For DSSP systems, a lower bound for the mean interfiring time $\Gamma^{(j)}$ of transition t_j (or its inverse an upper bound for the throughput σ_j^*) can be computed by solving the following linear programming problem:

$$\Gamma^{(j)} \geq \begin{array}{ll} \text{maximum} & Y^T \cdot PRE \cdot \vec{D}^{(j)} \\ \text{subject to} & Y^T \cdot C = 0 \\ & Y^T \cdot M_0 = 1 \\ & Y \geq 0 \end{array} \qquad \text{(LPP1)}$$

where PRE and C are matrices representing the Pre and global incidence functions of the net, M_0 is the initial marking, and $\vec{D}^{(j)}$ is the vector of average service demands for transitions.

We remark that the computation of the above bound has polynomial time complexity on the net size. This is because the computation of vector $\vec{D}^{(j)}$ is polynomial and because linear programming problems can also be solved in polynomial time.

If the solution of (LPP1) is unbounded and since it is a lower bound for the mean interfiring time of transition t_j, the non-liveness can be assured (infinite interfiring time). If the visit ratios of all transitions are non-null, the unboundedness of the problem (LPP1) implies that a total deadlock is reached by the net. This result has the following interpretation: if (LPP1) is unbounded then there exists an unmarked P-semiflow, and the system is non-live.

Concerning throughput lower bounds, provided the net system is live, they can be derived by adding the service time of all transitions, weighted by the visit ratios. This computation implies a complete sequentialization of all the activities represented in the model.

THEOREM 9 [8] *For well-behaved DSSP systems, an upper bound for the mean interfiring time* $\Gamma^{(j)}$ *of transition* t_j *(or its inverse a lower bound for the throughput) is:*

$$\Gamma^{(j)} \leq \sum_{i=1}^{m} s_i \, v_i^{(j)} = \sum_{i=1}^{m} D_i^{(j)} \tag{11}$$

where s_i, $v_i^{(j)}$, *and* $D_i^{(j)}$ *are the mean service time, visit ratio, and average service demand, respectively, for transition* t_i, $i = 1, \ldots, m$.

We remark that the above bound, provided the system is well-behaved, can also be computed in polynomial time, since the vector of visit ratios can be computed with such complexity.

Bounds for other performance indices can be computed using classical formulas in Queuing Networks theory such as Little's formula.

The number of tokens in a place defines the length of the represented queue (including the customers in service!). Thus it may be important to know bounds on average marking of places.

As an example, in [5] it has been shown that the following are lower and upper bounds for the average marking, $\overline{M}$:

$$\overline{M}^{lb} = PRE \cdot \mathcal{S} \cdot \vec{\sigma^*}^{lb} \tag{12}$$

$$\overline{M}^{ub}(p) \quad = \quad \max \{ \, M(p) \mid B^T \cdot M = B^T \cdot M_0 \, , \; M \geq \overline{M}^{lb} \, \} \qquad \text{(LPP2)}$$

where $\mathcal{S} = \text{diag}\,(s_i)$, $\vec{\sigma^*}^{lb}$ is the vector of throughput lower bounds, and the rows of B^T are the basis of left annullers of C (the incidence matrix of the net).

As an interesting remark, the reader can check that a structural absolute bound for the marking of a place is given for conservative nets (i.e., $\exists Y > 0$, $Y^T \cdot C = 0$) by the following expression:

$$SB(p) \quad = \quad \max \{ \, M(p) \mid B^T \cdot M = B^T \cdot M_0 \, , \; M \geq 0 \, \} \qquad \text{(LPP3)}$$

The constraint in (LPP3) being weaker than that in (LPP2) ($M \geq \overline{M}^{lb}$ is transformed into $M \geq 0$), it is obvious that $\overline{M}^{ub} \leq SB(p)$.

5 CONCLUSIONS AND FUTURE WORK

We have introduced a new structured subclass of Petri nets which generalizes the Deterministic Systems of Sequential Processes (DSSP) introduced in [18]. This class permits the modelling of cooperating sequential processes: the processes are modelled by safe State Machines while their cooperation is represented by places called buffers. The output-private and the deterministic assumptions, together with safeness of the State Machines, preclude competition.

We have presented several results concerning the functional and performance analysis of these models, and we have outlined some extensions and conjectures. The conjectures can be re-stated as follows:

- The potential reachability graph of a live DSSP is directed
- a DSSP is well-formed iff it is well-structured and $\text{rank}(C) = |\mathcal{E}| - 1$

The first one implies existence of home states in well-behaved systems and the characterization of liveness by the absence of solutions to a system of linear inequalities over the integers. The second is related to decomposability concepts and a compositional characterization of liveness. At present we are working out these topics.

REFERENCES

1. M. Ajmone-Marsan, G. Balbo, and G. Conte. A class of generalized stochastic Petri nets for the performance evaluation of multiprocessor systems. *ACM Transactions on Computer Systems*, 2(2):93-122, 1984.
2. M. Ajmone-Marsan, G. Balbo, and G. Conte. *Performance Models of Multiprocessor Systems.* MIT Press, 1986.
3. G.W. Brams. *Réseaux de Petri: Théorie et Pratique.* Masson, 1983.
4. J.M. Colom, J. Campos, and M. Silva. On liveness analysis through linear algebraic techniques. In *Procs. of the Annual General Meeting of ESPRIT Basic Research Action 3148 Design Methods Based on Nets (DEMON)*, Paris, June 1990.
5. J. Campos, G. Chiola, and M. Silva. Properties and performance bounds for closed Free Choice synchronized monoclass queuing networks. *IEEE Trans. Automatic Control*, 36(12):1368–1382, 1991.
6. J. Campos and M. Silva. Steady-state performance evaluation of totally open systems of Markovian sequential processes. In M. Cosnard and C. Girault (ed.) *Decentralized Systems*, 427-438. North-Holland, 1990.
7. J.M. Colom and M. Silva. Convex geometry and semiflows in P/T nets. A comparative study of algorithms for computation of minimal P-semiflows. In G. Rozenberg (ed.) *Advances in Petri Nets 1990*, volume 483 of *LNCS*, 77-112. Springer-Verlag, 1991.
8. J. Campos and M. Silva. Structural techniques and performance bounds of Stochastic Petri Net models. In G. Rozenberg (ed.) *Advances in Petri Nets 1992*, volume 609 of *LNCS*, 352-391. Springer-Verlag, 1992.
9. E. Gelenbe and G. Pujolle. *Introduction to Queuing Networks.* Wiley, 1987.
10. U. Herzog. Performance evaluation and formal description. In *Procs. of the IEEE Conference CompEuro 91*, 750-756, Bologna, Italy, May 1991.
11. C.A.R. Hoare. *Communicating Sequential Processes.* Prentice Hall, Englewood Cliffs, NJ, 1985.
12. R. Milner. *A Calculus of Communicating Systems.* Prentice Hall, London, 1989.
13. J. Martínez and M. Silva. A simple and fast algorithm to obtain all invariants of a generalized Petri net. In C. Girault and W. Reisig (ed.) *Application*

and Theory of Petri Nets, volume 52 of *Informatik-Fachberichte*, 301-310. Springer-Verlag, 1982.

14. T. Murata. Petri nets: Properties, analysis, and applications. *Proceedings of the IEEE*, 77(4):541-580, 1989.
15. W. Reisig. Deterministic buffer synchronization of sequential processes. *Acta Informatica*, 18, 117-134, 1982.
16. M. Silva. *Las Redes de Petri en la Automática y la Informática.* AC, 1985.
17. M. Silva. Interleaving functional and performance structural analysis of net models. In M. Ajmone Marsan (ed.) *Application and Theory of Petri Nets 1993*, volume 691 of *LNCS*, 17-23. Springer-Verlag, 1993.
18. Y. Souissi and N. Beldiceanu. Deterministic Systems of Sequential Processes: Theory and tools. In F.H. Vogt (ed.) *Concurrency 88*, volume 335 of *LNCS*, 380-400. Springer-Verlag, 1988.
19. Y. Souissi. Deterministic Systems of Sequential Processes: A class of structured Petri nets. In G. Rozenberg (ed.) *Advances in Petri Nets 1993*, volume 674 of *LNCS*, 406-426. Springer-Verlag, 1993.
20. E. Teruel, J.M. Colom and M. Silva. Linear analysis of deadlock-freeness of Petri net models. In J.W. Nieuwenhuis et al. (ed.) *Second European Control Conference*, volume 2, 513-518. North-Holland, 1993.
21. E. Teruel, J.M. Colom and M. Silva. Structure theory of Choice-free systems. Research Report GISI-RR-93-09, Dpto. Ing. Elect. Inform. Univ. Zaragoza, 1993. (Short version available in *Procs. of the ICDDS '93*, appearing with the title "Modelling and analysis of deterministic concurrent systems with bulk services and arrivals".)
22. E. Teruel and M. Silva. Liveness and home states in Equal Conflict systems. In M. Ajmone Marsan (ed.) *Application and Theory of Petri Nets 1993*, volume 691 of *LNCS*, 415-432. Springer-Verlag, 1993.
23. E. Teruel et al. Work in progress.

Stationary Regime and Stability of Free-Choice Petri Nets*

F. Baccelli and B. Gaujal
INRIA Sophia-Antipolis,
BP 93, 06902 Sophia-Antipolis, France

The aim of this paper is to give conditions under which a class of stochastic Petri nets called free choice nets satisfies a set of monotonicity and separability conditions ensuring the existence of a finite stationary regime for the marking process (or a finite periodic regime in the deterministic case). The scheme of conflict resolution is via a stochastic routing sequence. This assumption is essential for ensuring basic monotonicity properties. The main tool for proving these properties is ergodic theory.

1 The Separable-Monotone Framework for Counters and Daters

We consider a discrete event system composed of a set of nodes, and submitted to an input point process $N = \{T_n\}_{n\in\mathbb{Z}}$. Let $c.N$ be the c-dilation of N, namely the point process with arrivals $c.N = \{cT_n\}_{n\in\mathbb{Z}}$. Let $N_{[m,n]}$ be the $[m,n]$-restriction of the point process N, namely the point process $\{T_l\}_{m\le l\le n}$. We will say that $N_{[m,n]} \le N'_{[m,n]}$ if $T_l \le T'_l$ for all $m \le l \le n$. In what follows, this point process will also be characterized through its counting measure $R_{[m,n]} : \mathbb{R} \to \mathbb{N}$, where $R_{[m,n]}(t)$ counts the number of points of $N_{[m,n]}$ which are less than t.

The discrete event system is characterized by two equivalent sets of variables:

- The daters: $\mathcal{X}^i_{[m,n]}(k) \in \mathbb{R}$ will denote the epoch of the k-th event on node i, when the system is submitted to $N_{[m,n]}$ (here, we take $k \in \mathbb{N}$ and $\mathcal{X}^i_{[m,n]}(k) = \infty$ if there are less than k events on node i).
- The counters: $X^i_{[m,n]}(t) \in \mathbb{N}$ will denote the number of events which took place on node i before time t (we will take this function left-continuous).

Note that counters and daters are related by

$$X^i_{[m,n]}(t) = \sum_{k\in\mathbb{N}} 1_{\{\mathcal{X}^i_{[m,n]}(k)\le t\}} \cdot \tag{1}$$

The separable-monotone framework consists of the following set of assumptions:

*Supported by the European Grant BRA-QMIPS of CEC DG XIII. Also presented at the 11th International Conference on Analysis and Optimization of Systems, DES'94.

External Monotonicity If $N_{[m,n]} \leq N'_{[m,n]}$, then for all k and i (with obvious notations), $\mathcal{X}^i_{[m,n]}(k) \leq \{\mathcal{X}'\}^i_{[m,n]}(k)$, which is equivalent to the property that for all t and i $X^i_{[m,n]}(t) \geq \{X'\}^i_{[m,n]}(t)$.

Conservation Let

$$X^i_{[m,n]} \equiv \lim_{t\to\infty} X^i_{[m,n]}(t)\,. \tag{2}$$

This limit exists since the function is non-decreasing. In words, $X^i_{[m,n]}$ counts the *total* number of events on node i for $N_{[m,n]}$. We assume that $X^i_{[m,n]}$ is *finite* and *independent* of the values taken by the variables T_l, $n \leq l \leq m$ (provided m, n and $\{T_l\}$ are finite of course). Of particular interest to us will be the *maximal dater* defined by:

$$\mathcal{X}_{[m,n]} = \max_i \mathcal{X}^i_{[m,n]}(X^i_{[m,n]})\,. \tag{3}$$

Separability The separability assumption states that if $T_{l+1} \geq \mathcal{X}_{[m,l]} + M$, for some non-negative M, then

$$\begin{aligned} \mathcal{X}^i_{[m,n]}(k) &= \mathcal{X}^i_{[m,l]}(k), \quad k \leq X^i[m,l] \\ \mathcal{X}^i_{[m,n]}(k + X^i_{[m,l]}) &= \mathcal{X}^i_{[l+1,m]}(k), \quad k \geq 1 \end{aligned} \tag{4}$$

or equivalently

$$\begin{aligned} X^i_{[m,n]}(t) &= X^i_{[m,l]}(t), \quad t < T_{l+1} \\ X^i_{[m,n]}(t) &= X^i_{[m,l]} + X^i_{[l+1,m]}(t), \quad t \geq T_{l+1}. \end{aligned} \tag{5}$$

It is easy to check that the separation and the conservation properties imply that for all $m \leq l < n$, $X^i_{[m,n]} = X^i_{[m,l]} + X^i_{[l+1,n]}$ regardless of $\{T_l\}$.

Homogeneity The homogeneity assumption states that if $T'_l = T_l + c$, then $\{\mathcal{X}'\}^i_{[m,n]}(k) = \mathcal{X}^i_{[m,n]}(k) + c$ for all k and i or equivalently that $\{X'\}^i_{[m,n]}(t + c) = X^i_{[m,n]}(t)$ for all t and i.

Let

$$\begin{aligned} \mathcal{W}_{[m,n]} &\equiv \mathcal{X}_{[m,n]} - T_n, \\ W^i_{[m,n]}(t) &\equiv X^i_{[m,n]} - X^i_{[m,n]}(T_n + t), \quad t \geq 0\,. \end{aligned}$$

The following theorems are proved in [4]:

THEOREM 1 *Under the above properties, for all n, $\mathcal{W}_{[m-1,n]} \geq \mathcal{W}_{[m,n]}$; for all $t \geq 0$, i and n, $W^i_{[m-1,n]}(t) \geq W^i_{[m,n]}(t)$, so that*

$$\exists \lim_{m\to-\infty} \uparrow \mathcal{W}_{[m,n]} \equiv \mathcal{W}_{[-\infty,n]}, \quad \exists \lim_{m\to-\infty} \uparrow W^i_{[m,n]}(t) \equiv W^i_{[-\infty,n]}(t) \tag{6}$$

where the notation $\lim_{m\to-\infty} \uparrow x(m)$ implies that $x(m)$ is a non-decreasing function of m.

THEOREM 2 *If the system has stationary ergodic input point process defined on some probability space* $(\Omega, \mathcal{F}, P, \theta)$, *where* θ *is a shift on* Ω *which is ergodic and leaves* P *invariant, and if it is such that* $\mathcal{W}_{[n,n+k]} = \mathcal{W}_{[0,k]} \circ \theta^n$, *then the following a.s. limit takes place for all* $c \geq 0$*:*

$$\exists \lim_{n\to\infty} \frac{\mathcal{W}_{[0,n]}(c.N)}{n} = \gamma_c \tag{7}$$

where γ_c *is a constant. If the intensity* λ *of the input point process is such that* $\lambda\gamma_0 < 1$, *then* $\mathcal{W}_{[-\infty,n]}(1.N) \equiv \mathcal{W}_{[-\infty,n]} < \infty$, *for all* n. *If in addition* $W^i_{[n,n+k]}(t) = W^i_{[0,k]}(t) \circ \theta^n$ *for all* n, i *and* t *and*

$$\left\{\mathcal{W}_{[-n,0]} \to_{n\to\infty} \infty\right\} \overset{a.s.}{=} \left\{\exists i / W^i_{[-n,0]} \to_{n\to\infty} \infty\right\} \tag{8}$$

then, if $\lambda\gamma_0 < 1$, $W^i_{[-\infty,n]}(t) < \infty$ *for all* n, i *and* t.

Remark Often, one also defines classes of events like $\mathcal{X}^{i,j}(k)$, which count departures from node i to node j, and the above framework extends naturally to this type of variables (and the associated counters). In this case, it is natural to define the variables:

$$Q^i_{[m,n]}(t) = W^i_{[m,n]}(t) - \sum_j W^{j,i}_{[m,n]}(t) \tag{9}$$

which represent the number of objects (customers, tokens) on node i at time $t + T_n$ for $N_{[m,n]}$. So, if $\lambda\gamma(0) < 1$, we have constructed a stationary (θ-compatible) version of the $Q^i(t)$ process. For instance, $Q^i_{[-\infty,n]}(t)$ for $t \in [T_n, T_{n+1})$ provides a stationary process Q for the $[-\infty, +\infty]$-restriction of N, namely N.

2 TIMED PETRI NET

A Petri net is a t-uple $(\mathcal{P}, \mathcal{T}, \mathcal{C}, \mathcal{M}_0)$ where $\mathcal{P}$ is the set of places, $\mathcal{T}$ is the set of transitions, $\mathcal{C}$ the set of arcs between places and transitions and between transitions and places ($\mathcal{C}$ is a subset of $\mathcal{P} \times \mathcal{T} \cup \mathcal{T} \times \mathcal{P}$). $\mathcal{M}_0$ is the initial marking in the places. We denote by ${}^\bullet t$ the set $\{p \in \mathcal{P} : (p,t) \in \mathcal{C}\}$ (i.e. the set of all the input places of t). We define similarly the sets $t^\bullet$, ${}^\bullet p$, $p^\bullet$ as the set of the output places of t, the set of the input transitions of p and the set of the output transitions of p, respectively.

A timed Petri net is a Petri net with timings attached to transition firings: $\sigma^t(n)$ is the duration of the n-th firing of transition t. This means that if transition t begins to fire for the n-th time at epoch e, this firing will end at epoch $e + \sigma^t(n)$ and tokens are taken out of input places and put in output places of t according to the firing rule of a Petri net.

2.1 Free Choice Nets

Free choice nets (FCN) are Petri nets verifying the following conditions: $\forall p \in \mathcal{P}, t_1, t_2 \in p^\bullet, t_1 \neq t_2, \ {}^\bullet t_1 = {}^\bullet t_2 = \{p\}$. In other words, whenever two transitions share an input place, they have no other input place.

Free choice nets have been extensively studied in the 70's [6] and have regained interest recently [7], [9] because they constitute a nice compromise between power of description and tractability of problems.

For any place p with several output transitions, the dynamic is characterized by a routing function $\nu^p : I\!N \rightarrow p^\bullet$ associated with place p, where $\nu^p(k)$ is the transition to which the k-th token to enter place p is routed. The routing function can be a deterministic or a random sequence. For comments on this type of routing policy, see [1].

2.2 Decomposition into Marked Graphs Components

A place p in a FCN F is *serial* if $|{}^\bullet p| = |p^\bullet| = 1$.

First we define a relation $\mathcal{L}$ by: $t, t' \in T, t\mathcal{L}t'$ if there is a serial place p verifying $\{{}^\bullet p, p^\bullet\} = \{t, t'\}$. Let $\mathcal{K}$ be the transitive closure of $\mathcal{L}$. $\mathcal{K}$ is a parallelism relation. We partition the set of transitions $\mathcal{T}$ into its maximal $\mathcal{K}$-classes, $\mathcal{T}_1, \cdots, \mathcal{T}_n$. We construct a decomposition of F in the following way: $\mathcal{P}_i = \{p \in \mathcal{P} | p \text{ serial and } {}^\bullet p, p^\bullet \in \mathcal{T}_i\}$ for all $1 \leq i \leq n$.

The marked graph component (MGC) G_i of F is the sub-Petri net of F $(\mathcal{P}_i, \mathcal{T}_i, \mathcal{C} \cap (\mathcal{P}_i \times \mathcal{T}_i \cup \mathcal{T}_i \times \mathcal{P}_i))$. One can easily check that G_i is a marked graph and is maximal in the sense that no marked graph included in F contains G_i, except G_i itself.

The places which do not belong to any component G_i are the places with several input transitions and/or several output transitions. These places will be called *routing* places in the following. The set of the routing places is denoted $\mathcal{R}$.

2.3 Classification of Free Choice Nets

We propose a classification of the marked graph components of a FCN based upon its links with the routing places.

A MGC G_i is said *Single Input* (SI) if $\#\{t \in \mathcal{T}_i, {}^\bullet t \not\subset \mathcal{P}_i\} = 1$. G_i is said *Multiple Input* (MI) if $\#\{t \in \mathcal{T}_i, {}^\bullet t \not\subset \mathcal{P}_i\} > 1$. A MGC G_i is said *Single Output* (SO) if $\#\{t \in \mathcal{T}_i, t^\bullet \not\subset \mathcal{P}_i\} = 1$. A MGC G_i is said *Multiple Output* (MO) if $\#\{t \in \mathcal{T}_i, t^\bullet \not\subset \mathcal{P}_i\} > 1$.

Thus all the MGC of a FCN can be put in one of the four classes, SISO, SIMO, MISO, MIMO. A FCN is said SI (resp. SO, MI, MO , SISO , SIMO, MISO, MIMO) if all its MGC are SI (resp. SO, MI, MO , SISO, SIMO, MISO, MIMO).

3 Evolution Equations for Counters

Consider a FC net satisfying the following assumption: a transition t with more than one incoming arc (i.e. with an and-convergence) is never preceded by a place p with more than one incoming arc (i.e. with an or-convergence). This restriction introduces no loss of generality: because of the FC constraint, a transition t as above cannot be preceded by a place with multiple outcoming arcs; in addition, each place p as above can be replaced by a triple p', t', p'',

where ${}^\bullet p' = {}^\bullet p$, $p''^\bullet = p^\bullet = t$, and where $p'^\bullet = t'$, ${}^\bullet t' = p'$, $t'^\bullet = p''$, ${}^\bullet p'' = t'$, without altering the evolution of the net.

Let $X^t(u)$ denote the counter associated with t, namely, the number of firings initiated by transition t by time u. We will consider the version of this process that is continuous to the right. Let $\mathcal{A}$ be the set of transitions such that all their upstream places are serial, and let $\mathcal{B}$ be the set of transitions which do not belong to $\mathcal{A}$. Let $Y(u)$ be the vector $\{X^t(u),\ t \in \mathcal{A}\}$, where the transitions are arranged in some order, and let $Z(u)$ be the vector $\{X^t(u),\ t \in \mathcal{B}\}$. Note that each transition of $\mathcal{B}$ has at most one non-serial upstream place due to the FC constraint. However, this place may precede several transitions of $\mathcal{B}$.

Constant Firing Times We shall first consider the case when firing times are constant, positive, and all multiple of a common number, which will be taken equal to 1 without loss of generality. These assumptions are essentially for the sake of easy exposition. We will in particular show in Section 6 how to address the case with stochastic times, which can be treated with a similar method. We will denote M the (integer-valued) upper bound on the firing times. We will denote $\nu^p(m)$ the m-th routing decision from place p ($\nu^p(m) \in p^\bullet$) and $\Pi^t(m)$ the sum

$$\Pi^t(m) = \sum_{l=1}^{m} 1_{\nu^{\bullet t}(l)=t}, \quad t \in \mathcal{B}. \tag{10}$$

Similarly, for sake of easy exposition, we will limit ourselves to the case when the jump times T_n of R are integer-valued.

LEMMA 1 *Under the above assumptions, for all integers n, the counting vectors $\{Y(k), Z(k)\}$ satisfy the following evolution equation, which is valid for $n > M$:*

$$Y(k) = \bigoplus_{l=1}^{M} (A_l \otimes Y(k-l) \oplus B_l \otimes Z(k-l)) \tag{11}$$

$$Z(k) = \Pi\left(\sum_{l=1}^{M} (P_l \times Z(k-l) + Q_l \times Y(k-l)) + R(k)\right), \tag{12}$$

where $(\oplus, \otimes)$ is $(\min, +)$ (see [2]), $(+, \times)$ is the usual algebra, and

- *The matrix A_l on $\mathcal{A} \times \mathcal{A}$ is defined by $A_l(t, t') = c$ if the firing time of $t \in \mathcal{A}$ is l and there is a serial place with c initial tokens between $t' \in \mathcal{A}$ and t; ∞ otherwise. If there are more than one serial place between t' and t, we take c equal to the minimum of their initial markings.*
- *The matrix B_l on $\mathcal{A} \times \mathcal{B}$ is defined by $B_l(t, t') = c$ if the firing time of t is l and there is a serial place with c initial tokens between $t' \in \mathcal{B}$ and $t \in \mathcal{A}$; ∞ otherwise.*
- *The matrix P_l on $\mathcal{B} \times \mathcal{B}$ is defined by $P_l(t, t') = 1$ if the firing time of $t \in \mathcal{B}$ is l and there is a place connecting t' to t; 0 otherwise.*

- *The matrix Q_l on $\mathcal{B} \times \mathcal{A}$ is defined by $Q_l(t,t') = 1$ if the firing time of $t \in \mathcal{B}$ is l and there is a place connecting t' to t; 0 otherwise.*
- *$R(0)$ is the matrix of initial markings: $R(t',t) = c$ if $t \in \mathcal{B}$ is such that ${}^\bullet t$ has an initial marking of c. More generally, $R(k)(t',t)$ is the cumulated external input in that place up to time n. $R(k)(t',t) = R(k-1)(t',t) + I(k)(t,t')$. (thus $I(k) \neq 0$ iff $k \in \{T_n\}_n$).*
- *For all vectors of integers $Z = (Z_1, \ldots, Z_q)$, where $q = |\mathcal{B}|$, $\Pi(Z)$ is the vector of integers $(\pi_1(Z_1), \ldots, \pi_q(Z_q))$.*

Proof Immediate from definitions, the key observation being that due to our preliminary assumption, a transition which belongs to $\mathcal{B}$ will never have more than one input arc, which allows us to write (12). ■

Total Number of Firings Let $Z = Z(\infty)$ and $Y = Y(\infty)$ denote the vectors counting the total number of firings of the transitions. One can easily check the absence of deadlocks (a deadlock is a marking where no transition can fire) and related properties, from Y and Z: for instance, the system is deadlocked for the initial marking $(\forall k,\ R(k) = R(0))$ if and only if Z and Y are finite.

LEMMA 2 *The integer-valued vectors Z and Y satisfy the system of equations*

$$Y = A \otimes Y \oplus B \otimes Z \tag{13}$$

$$Z = \Pi(P \times Z + Q \times Y + R), \tag{14}$$

where $A = \bigoplus_{l=1}^{M} A_l$, $B = \bigoplus_{l=1}^{M} B_l$, $P = \sum_{l=1}^{M} P_l$, $Q = \sum_{l=1}^{M} Q_l$, $R = \lim_{k\to\infty} R(k)$ are independent of n.

A notable property is that this system does not depend on the variables σ^t anymore: in other words, all properties like liveness, deadlock and intermediates are associated with the switching functions, the topology and the initial marking only, and not with timing variables.

3.1 Assumptions on the Function Π

We assume that at the origin of time, the network is in a configuration where no transition can fire. Such a marking is called the *original deadlock.*

In the following, we add two assumptions on the function Π.

ASSUMPTION 1 (A_1) *The total firing vectors Z and Y are finite if R is finite.*

This assumption is a property of the function Π since the total firing vector does not depend on the timing variables σ^t of the system. This assumption says that if the number of arrivals is finite, the system reaches a deadlock after a finite number of firings.

ASSUMPTION 2 (A_2) *If the network reaches a deadlock, then this deadlock is the original deadlock.*

We can give a characterization of this property in terms of the vector $X = (Z, Y)$. If the network reaches a deadlock, then X is finite. If this is the original deadlock, then in any MGC G, all the transitions have fired the same number of times, $\forall t_1, t_2 \in G,\ X^{t_1} = X^{t_2}$. Conversely if X is finite and for all MGC G, $\forall t_1, t_2 \in G,\ X^{t_1} = X^{t_2}$, then the network has reached the original deadlock.

3.2 Restriction of the Arrival Process

Let $R^t_{[m,n]}$ be the counting measure of $N_{[m,n]}$ on place ${}^\bullet t$.

The assumption (A_1) allows us to say that the network with the input process $\left(R_{[0,n]}(k)\right)_{k\in I\!N}$ reaches a deadlock. We denote by $Z_{[0,n]}$ and $Y_{[0,n]}$ the total firing vectors for this system. With $\Pi_{[0,\infty]} \equiv \Pi$, they verify:

$$\begin{aligned} Y_{[0,n]} &= A \otimes Y_{[0,n]} \oplus B \otimes Z_{[0,n]} \\ Z_{[0,n]} &= \Pi_{[0,\infty]}\left(P \times Z_{[0,n]} + Q \times Y_{[0,n]} + R_{[0,n]}\right) . \end{aligned}$$

If $t \in \mathcal{B}$, we denote by $U^{{}^\bullet t}_{[0,n]}$ the total number of tokens that entered the place ${}^\bullet t$ and $U_{[0,n]} = \{U^{{}^\bullet t}_{[0,n]},\ t \in \mathcal{B}\}$. We have

$$U_{[0,n]} = P \times Z_{[0,n]} + Q \times Y_{[0,n]} + R_{[0,n]}. \tag{15}$$

Since $Z_{[0,n]}$ and $Y_{[0,n]}$ are finite, $U_{[0,n]}$ is also finite.

Now, we introduce the system generated by the restricted input process $(R_{[m,n]}(k))_{k\in I\!N}$. We connect this system with the original one by taking

$$\nu^p_{[m,\infty]}(k) \equiv \nu^p(k + U^t_{[0,m-1]}), \tag{16}$$

for all $t \in \mathcal{T}$ and for all $p \in \mathcal{R}$. Thus the function $\Pi_{[m,\infty]}$ is defined by

$$\Pi^t_{[m,\infty]}(k) \equiv \sum_{l=1}^{k} \mathbf{1}_{\nu^{{}^\bullet t}_{[m,\infty]}(l)=t} = \Pi^t(k + U^{{}^\bullet t}_{[0,m-1]}) - \Pi^t(U^{{}^\bullet t}_{[0,m-1]}) .$$

Finally we define the vectors $Y_{[m,n]}(k)$ and $Z_{[m,n]}(k)$, which verify the equations

$$\begin{aligned} Y_{[m,n]}(k) &= \bigoplus_{l=1}^{M} \left(A_l \otimes Y_{[m,n]}(k-l) \oplus B_l \otimes Z_{[m,n]}(k-l)\right) \\ Z_{[m,n]}(k) &= \Pi_{[m,\infty]}\Bigg(\sum_{l=1}^{M} \left(P_l Z_{[m,n]}(k-l) + Q_l Y_{[m,n]}(k-l)\right) + \\ &\qquad\qquad R_{[m,n]}(k)\Bigg), \end{aligned}$$

with the initial conditions : $\forall k < T_m$, $Y_{[m,n]}(k+1) = 0$ and $Z_{[m,n]}(k) = 0$.

4 The Saturation Rule

We prove that the variables $X_{[m,n]}(k)$ satisfy the extended saturation conditions. We need a preliminary lemma.

Lemma 3 *If* $T_m > \mathcal{X}_{[0,m-1]} + M$ *then for all* $k \geq T_m$, $X_{[0,n]}(k) = X_{[0,m-1]} + X_{[m,n]}(k)$.

Proof The proof holds by induction on k. For $k = T_m$, $X_{[1,m-1]} = X_{[1,m-1]}(k-l)\ \forall l > 1$ yields

$$\begin{aligned} Y_{[0,n]}(k) &= Y_{[0,m-1]} \\ Z_{[0,n]}(k) &= \Pi\left((PZ_{[0,m-1]} + QY_{[0,m-1]} + R_{[0,m-1]} + I(T_m)\right). \end{aligned}$$

So, by definition of $\Pi_{[m,\infty]}$:

$$\begin{aligned} Y_{[0,n]}(k) &= Y_{[0,m-1]} + Y_{[m,n]}(k) \\ Z_{[0,n]}(k) &= Z_{[0,m-1]} + Z_{[m,n]}(k). \end{aligned}$$

For the case $k > T_m$, the induction property yields

$$\begin{aligned} Y_{[0,n]}(k) &= \bigoplus_{l=1}^{M}(A_l \otimes (Y_{[0,m-1]} + Y_{[m,n]}(k-l)) \oplus B_l \otimes \\ &\qquad (Z_{[0,m-1]} + Z_{[m,n]}(k-l))) \\ Z_{[0,n]}(k) &= \Pi\Big(\sum_{l=1}^{M}\big(P_l(Z_{[0,m-1]} + Z_{[m,n]}(k-l)) \\ &\quad + Q_l(Y_{[0,m-1]} + Y_{[m,n]}(k-l))\big) + R_{[0,m-1]} + R_{[m,n]}(k)\Big). \end{aligned}$$

Using the characterization of assumption (A_2) for $Y_{[0,m-1]}$,

$$\begin{aligned} Y_{[0,n]}(k) &= Y_{[0,m-1]} + \bigoplus_{l=1}^{M}\left(A_l \otimes Y_{[m,n]}(k-l) \oplus B_l \otimes Z_{[m,n]}(k-l)\right) \\ Z_{[0,n]}(k) &= \Pi\Big(\sum_{l=1}^{M}(P_l Z_{[m,n]}(k-l) + Q_l Y_{[m,n]}(k-l)) + \\ &\qquad U_{[0,m-1]} + R_{[m,n]}(k)\Big). \end{aligned}$$

This yields

$$\begin{aligned} Y_{[0,n]}(k) &= Y_{[0,m-1]} + Y_{[m,n]}(k) \\ Z_{[0,n]}(k) &= Z_{[0,m-1]} + Z_{[m,n]}(k). \end{aligned}$$

■

Corollary 1 $X_{[0,n]} = X_{[0,m-1]} + X_{[m,n]}$.

Proof This is an immediate corollary of the previous lemma considering the fact that $X_{[0,n]}$ does not depend on T_m. ■

LEMMA 4 (EXTERNAL MONOTONICITY) *With two arrival counting measures* $R'_{[m,n]}(k) \geq R_{[m,n]}(k)$, *then* $X'_{[m,n]}(k) \geq X_{[m,n]}(k)$.

Proof The vector $X_{[m,n]}(k)$ is an increasing function of $\left(X_{[m,n]}(k-l)\right)_{l=1\cdots M}$ and of $R_{[m,n]}(k)$. The proof follows by a straightforward induction. ■

LEMMA 5 (CONSERVATION) $X_{[m,n]}$ *is finite and independent of the arrival times.*

Proof Corollary 1 says that $X_{[m,n]} = X_{[0,n]} - X_{[0,m-1]}$. Therefore, $X_{[m,n]}$ is finite and independent of the arrival times. ■

LEMMA 6 (SEPARABILITY) *Suppose* $T_r > \mathcal{W}_{[m,n]} + M$. *Then for all* $m \leq n$,
if $k < T_r$, *then* $X_{[m,n]}(k) = X_{[m,r-1]}(k)$,
if $k \geq T_r$, *then* $X_{[m,n]}(k) = X_{[m,r-1]} + X_{[r,n]}(k)$.

Proof The case $t < T_r$ is trivial. For the case $t \geq T_r$, the proof holds by induction on k. It is very similar to the proof of lemma 3 and is not reported here. ■

LEMMA 7 (HOMOGENEITY) *Let* $R'_{[m,n]}$ *be the arrival process shifted by a constant* C, $R'_{[m,n]}(k) = R_{[m,n]}(k+C)$. *Then,* $X'_{[m,n]}(k) = X_{[m,n]}(k+C)$.

Proof This holds by immediate induction on k. ■

4.1 Stochastic Assumptions

All the random variables defined in what follows are assumed to be carried by some probability space $(\Omega, \mathcal{F}, P, \theta)$, where θ is an ergodic shift which leaves P invariant. We assume that the point process associated with the counting measure $R_{[-\infty,+\infty]}(k)$ is θ-stationary and ergodic, and that it has a finite intensity. When taking $\{T_0 = 0\}$, this θ-stationarity assumption here means that

$$R_{[n,\infty]}(T_n + k) = R_{[0,\infty]}(k) \circ \theta^n \tag{17}$$

for all $k \in I\!R$ and n. Consider the $\mathcal{T}$-valued sequences $\nu_{[0,\infty]} = \{\nu^p_{[0,\infty]}(k)\}$, $p \in \mathcal{R}, k \in I\!N\}$ describing the routing decisions; we also assume that the following compatibility relation holds for all n.

$$\nu_{[n,\infty]} = \nu_{[0,\infty]} \circ \theta^n \,. \tag{18}$$

If $\mu = \{\mu(k)\}_{k\geq 0}$ denotes a sequence, for all integers $V \geq 0$, we will denote $\tau_V \mu$ the shifted sequence $\{\mu(V+k)\}_{k\geq 0}$. If $U = (U^p,\ p \in \mathcal{R})$ is a vector of non-negative integers, we will denote $\tau_U \nu$ the sequences $(\tau_{U^p}\nu^p,\ p \in \mathcal{R})$. Equation (16) and the above relation imply that

$$\nu_{[0,\infty]} \circ \theta^n = \tau_{U_{[0,n-1]}} \nu_{[0,\infty]} \tag{19}$$

where $U^{\bullet t}_{[0,n]}$ is the function defined in Equation (15).

It should be clear that under the above assumptions, the functions $\Pi^t_{[n,\infty]}$ satisfy the compatibility property

$$\Pi_{[n,\infty]} = \Pi_{[0,\infty]} \circ \theta^n \tag{20}$$

so that the compatibility relations of Theorem 2 hold for the counters and the daters of the FCN. Besides this, since all the firing times are positive, one can easily check that condition (8) is satisfied. Thus, Theorem 2 holds for this class of systems under the above assumptions.

Remark Note that since $\nu^p_{[0,\infty]}(k) = \nu^p_{[n,n]}(l)$ for $l = k - U^p_{[0,n-1]} \leq U^p_{[n,n]}$, the infinite sequences $\nu_{[0,\infty]}$ are fully determined by the finite sub-sequences

$$(\nu^p_{[n,n]}(k),\ p \in \mathcal{R},\ 0 \leq k \leq U^p_{[n,n]})_{n \geq 0}$$

Thus, with our framework, for all nodes p, the whole routing sequence $\nu^p_{[0,\infty]}(k)$ is simply the concatenation of the routing sequences $\nu^p_{[n,n]}(l)$, $1 \leq l \leq U^p_{[n,n]}$.

5 Analysis of an Example: The SI-FCN

The aim of this section is to give sufficient conditions for the assumptions of the preceding section to hold. We will limit ourselves to the SI-FCN case, where certain simplifications take place.

A MGC G_i is *input-connected* if for each transition in G_i, there is a path from its input transition to t. This is equivalent to: B has no lines the values of which are all $+\infty$.

Lemma 8 *Let F be a SI-FCN with all its MGC input-connected. If F can reach a deadlock, then this deadlock is unique.*

Proof If a routing place p contains a token, then one of the transitions in $p^\bullet$ is enabled, thus this marking is not a deadlock. Let t be a transition in G_i, let us follow the longest path in G_i without tokens. This path leads to a transition which is enabled except if it is the input transition. Now, a marking verifying these conditions is necessarily unique. ∎

Therefore, for SI-FCN with input-connected components, assumption (A_2) is redundant.

For all $t \in \mathcal{B} \cap G_i$, let O^t be the set

$$O^t = \{q \in \mathcal{R} \mid \exists t \in G_i \text{ s.t. } q = t^\bullet\},$$

where q is counted with multiplicity n if there are n arcs going from G_i to q. We will then say that q is an *offspring* of t with multiplicity n.

We now describe the dynamics of a pseudo marking process (this marking process is different from the one in the real system) on the set of places of $\mathcal{R}$,

which is driven by the routing functions only. Fix an arbitrary priority order on the nodes of $\mathcal{R}$: p_j has priority over p_{j+1} etc. Assume that the jump of R at T_0 brings $m^{\bullet t}$ tokens to place ${}^\bullet t$, for all $t \in \mathcal{B}$. If $m^{p_1} > 0$, one token of p_1 is *moved* following the routing decision $t = \nu^{p_1}_{[0,0]}(1)$. Let n^{p_i} be the multiplicity of the offspring p_i of t (this multiplicity is zero if p_i does not belong to O^t). By definition, such a move leads to the new marking defined by $m^{p_i} + n^{p_i}$ for all $i > 1$ and $m^{p_1} + n^{p_1} - 1$ for p_1. If the new marking of p_1 is still positive, we move one token of p_1 as above, but according to the routing decision $\nu^{p_1}_{[0,0]}(2)$, which leads to a new marking; the procedure is repeated up to the time when no tokens are left in p_1 (this may never happen in which case this first step of the procedure never stops). We then move one token of type p_2 according to the routing decision $\nu^{p_2}_{[0,0]}(1)$, provided there is at least one token in place p_2 in the last obtained marking. This may possibly create new tokens of type p_1. The general rule is actually to move the token with highest priority at each step, according to the residual routing decisions. The procedure stops whenever there are no tokens left in the routing places.

LEMMA 9 *The assumptions* (A_1) *(and therefore* (A_2)*) are satisfied if and only if the above procedure stops after an almost surely finite number of steps.*

The proof is omitted. It is based on a generalization of the Euler property for directed graphs called the Euler-Ordered property, which is introduced in [3]. Note that if the above stopping property holds for this specific ordering of the moves, it will hold for any other ordering.

In the particular case of i.i.d. routing decisions, independent on different nodes, one can naturally associate a multitype branching process with the set $\mathcal{R}$ by saying that an individual of type p has a set of offspring O^t with probability $P = P(\nu^p = t)$. Properties (A_1) (and (A_2)) will then a.s. hold whenever this multitype branching process is subcritical (namely whenever its population dies out a.s. for all finite initial conditions). This property boils down to checking that the maximal eigenvalue of the branching matrix is strictly less than 1 ([8]).

6 GENERALIZATION TO VARIABLE FIRING TIMES

Variable Firing Times We now consider the case when firing times are still integer-valued and bounded, but variable with time. Let $\sigma^t(m)$ be the firing time of the m-th firing of transition t. Let $\zeta^t(k)$ be the minimum of M and the time which elapsed since the last time t has started firing before time k. If we consider the variables to be left-continuous, we have:

LEMMA 10

$$Y(k) = \bigoplus_{l=1}^{M} (A_l(k) \otimes Y(k-l) \oplus B_l(k) \otimes Z(k-l)) \tag{21}$$

$$Z(k) = \Pi\left(\sum_{l=1}^{M} (P_l(k) \times Z(k-l) + Q_l(k) \times Y(k-l)) + R(k)\right), \tag{22}$$

with $A_l(k)(t,t') = c$, the number of tokens in the initial marking of the place between t' and t if $\zeta^t(k) = l$, ∞ otherwise (with a similar definition for B) and $P_l(k)(t,t') = c$, the number of tokens in the initial marking of the place between t' and t if $\zeta^t(k) = l$, 0 otherwise (with a similar definition for Q).

A system with variable firing times also falls within the monotone separable framework and the same method as in the constant case can be applied (see [5]).

6.1 Future Research

The constant γ_0 was computed for Jackson networks ([3]) which happen to be a simple case of SI-FCN. Its computation within the more general framework of this paper is considered in [5]. Other results can also be obtained along the lines of [3] including conditions for coupling convergence with (and uniqueness of) the stationary regime.

REFERENCES

1. Baccelli F., Cohen G., Gaujal B., *Recursive Equations and Basic Properties of Timed Petri Nets*, DEDS: Theory and Applications, 1(4)415-439, 1992.
2. Baccelli F., Cohen G., Olsder G.J, Quadrat J.P. *Synchronization and Linearity*, Wiley 1992.
3. Baccelli F., Foss S., *Stability of Jackson-type Queueing Networks, I*, INRIA Report n. 1845, To appear in *Queueing Systems.*
4. Baccelli F., Foss S., *On the Saturation Rule for the Stability of Queues*, INRIA Report n. 2015, 1993.
5. Baccelli F., Foss S., Gaujal B., *On the stability region for a class of stochastic Petri nets*, in preparation.
6. Commoner F., *Deadlocks in Petri Nets*, Applied Data Research, CA-7606-2311, 1972.
7. Esparza L., Silva M., *On the Analysis and Synthesis of Free Choice Systems*, G. Rozenberg editor, Advances in Petri Nets, Vol. 483 of LNCS, 243-286, 1990.
8. Harris T.E., *The Theory of Branching Processes,* Springer, Berlin, 1969.
9. Thiagarajan P.S., Voss K., *A Fresh Look at Free Choice Nets*, Information and Control, (62)85-113, 1984.

A General Iterative Technique for Approximate Throughput Computation of Stochastic Marked Graphs

J. Campos*, J.M. Colom*, H. Jungnitz†, and M. Silva*
Departamento de Ingeniería Eléctrica e Informática
Centro Politécnico Superior, Universidad de Zaragoza
María de Luna, 3, 50015 Zaragoza, Spain

A general iterative technique for approximate throughput computation of stochastic strongly connected marked graphs is presented. It generalizes a previous technique based on net decomposition through a single input-single output cut, allowing the split of the model through any cut. The approach has two basic foundations. First, a deep understanding of the qualitative behaviour of marked graphs leads to a general decomposition technique. Second, after the decomposition phase, an iterative response time approximation method is applied for the computation of the throughput. Experimental results on several examples generally have an error of less than 3%. The state space is usually reduced by more than one order of magnitude; therefore the analysis of otherwise intractable systems is possible.

1 INTRODUCTION

Stochastic Marked Graphs (SMG's) are a well-known subclass of stochastic Petri net models. They allow concurrency and synchronization but not decisions. From a queueing network perspective, it can be seen [13] that, provided strong connectivity, they are isomorphic to *fork/join queueing networks with blocking* (FJQN/B).

In this paper we consider strongly connected SMG's with time and marking independent *exponentially distributed* service times associated with transitions. For this class of models, several computation techniques have been presented in the literature. Exact performance results can be obtained from the numerical solution of the underlying continuous time Markov chain (CTMC) [3], but the *state explosion problem* makes intractable the evaluation of large systems. The efficient computation of exact performance indices of SMG's cannot be done analytically because *local balance property* does not hold in general [15]. The alternative approach of *bounds computation* has been studied by several authors using different techniques (see, e.g., [5, 8]).

*This work was partially supported by the European ESPRIT BRA Project 7269 QMIPS, the Spanish PRONTIC 354/91, and the Aragonese CONAI-DGA P-IT 6/91.

†The work by this author was performed while he was visiting the University of Zaragoza

Concerning approximation techniques, several proposals have been done. In [4], a method is proposed for nets that admit a *time scale decomposition* based on *near-complete decomposability* of Markov chains. Near decomposability properties are also used in [10] for an iterative approximate solution of weakly connected nets. In [6], some particular queueing networks with subnetworks having *population constraints* are analyzed using *flow equivalent aggregation* (i.e., a non-iterative technique) and Marie's method [20] (the idea is to replace a subsystem by an equivalent exponential service station with load-dependent service rates obtained by analyzing the subsystem in isolation under a load-dependent Poisson arrival process). An alternative approach is presented in [18] to compute approximate throughput for SMG's. In that work, the original system is also *split in subsystems* and a *delay equivalence* criterion is used for throughput approximation. The service rates for the aggregated subsystems are *marking dependent*. In [16], *response time approximation* is applied for an iterative computation of the throughput of SMG's. The main differences with respect to the work in [18] are two: first, the splitting of the MG is more firmly based on *qualitative theory* of MG's and second, the service rates for aggregated subsystems are *constant* (similar accuracy of the throughput is obtained with simpler and more robust algorithms).

A discussion of the above recalled techniques is presented in [17] for the throughput approximation of SMG's. We summarize now some of the conclusions. Flow equivalent aggregation is clearly the most efficient method (it is not an iterative method). In this method, the behaviour of the subsystem is assumed to be independent of the arrival process and depends only on the number of customers in the system. In many cases, this assumption is violated (see [16]), therefore the method cannot be applied.

Marie's method behaves correctly in many cases. As with many iterative methods, the uniqueness of the solution cannot be proven although numerical experience has shown that a unique point does indeed exist. The main drawback is that convergence sometimes presents a problem [7].

Concerning the delay equivalence technique presented in [18], its convergence may sometimes constitute a problem. The robustness of the method is improved in [19], where the service rates of the aggregated subsystems are made constant. Some problems of this approach have been reported in [16] where it is shown that the speed of convergence strongly depends on the initial values estimated for the service rates that represent the aggregated subsystem. Moreover, for several models the authors were not able to find initial values for which the method would converge.

Finally, the response time approximation method introduced in [16] shows similar accuracy as delay equivalence, but at a greatly reduced computational cost. The method seems to be insensitive with respect to the initial values of service rates and is the one which requires the least amount of iterations. The main drawback of this method is that the original MG must be decomposed into two subsystems each one with only one input place and only one output place (*single input-single output* or *SISO cut*), and such decomposition is not always possible. A generalization to *SIMO* (single input-multiple output), *MISO*

(multiple input-single output), and *MIMO* (multiple input-multiple output) cuts has been proposed but it presents serious problems concerning the quality of the results [17]. These problems are due to the fact that the structure of the net must be modified, by adding a "dummy synchronization" transition, to get a SISO cut, and this transformation can lead to a system with a considerably different behaviour.

In this paper, we follow an *iterative response time approximation technique* that avoids the problems derived from the application of the method in [16] for the general cases (SIMO, MISO, or MIMO cuts). The approach is deeply based on *qualitative theory of MG's.* More precisely, given an arbitrary cut (subset of places producing a net partition), a *structural decomposition* technique is developed in this paper that allows us to split a strongly connected MG into two *aggregated subsystems* and a *basic skeleton system.* And what is more important, *the behaviours of the subsystems, including steps, language of firing sequences and reachable markings, are equivalent to the whole system behaviour* (projected on the corresponding subsets of nodes). The better the qualitative behaviour of the system is represented by the aggregated subsystems, the more accurate the quantitative approximation will be.

The paper is organized as follows. In Section 2, basic notation and fundamental properties on MG's and implicit places are presented. Section 3 includes the structural decomposition of MG's used in the rest of the paper. The iterative technique for approximate throughput computation is described in Section 4. Section 5 includes several application examples to illustrate the introduced technique. Finally, concluding remarks are presented in Section 6.

2 Basics on Stochastic Marked Graphs

2.1 Basic notations

We assume that the reader is familiar with concepts of P/T nets. In this section we present notations used in later sections, for further extensions the reader is referred to [21, 22].

$\mathcal{N} = (P, T, F)$ is a net if P and T are disjoint sets of places and transitions, respectively and $F \subseteq (P \times T) \cup (T \times P)$. We shall only consider nets with finite and nonempty sets of places and transitions. A net is connected if and only if the least equivalence relation which includes F is $(P \cup T) \times (P \cup T)$.

Let $\mathcal{N} = (P, T, F)$ be a net. A path of $\mathcal{N}$ is a sequence $x_1 \ldots x_k$ of elements (places and transitions) of $\mathcal{N}$ satisfying $(x_1, x_2), \ldots, (x_{k-1}, x_k) \in F$. It is a circuit if $(x_k, x_1) \in F$. A path (circuit) is called simple if all elements in the sequence defining the path (circuit) are different. In this paper we only consider simple paths and circuits. We denote by $\mathcal{P}(x, y)$, $x, y \in P \cup T$, the set of simple paths from x to y. This notion is extended to sets of elements: $\mathcal{P}(X, Y)$ is the union of the $\mathcal{P}(x, y)$ for all $x \in X$ and for all $y \in Y$.

$\mathcal{N}$ is strongly connected if for every two elements x, y of $\mathcal{N}$ there exists a path $x \ldots y$. Pre- and Post-sets of elements are denoted by the dot-notation: ${}^\bullet x = \{y | (y, x) \in F\}$ and $x^\bullet = \{y | (x, y) \in F\}$. This notion is extended to sets of elements; ${}^\bullet X$ is the union of the pre-sets of elements of X, $X^\bullet$ is the union

of the post-sets of elements of X.

A function $M : P \rightarrow \{0, 1, \ldots\}$ (usually represented in vector form) is called a marking. A net system is a couple $\langle \mathcal{N}, M_0 \rangle$ of a net $\mathcal{N}$ and an initial marking M_0. A transition t is enabled at marking M if for all $p \in {}^\bullet t, M[p] > 0$. An enabled transition can be fired. The fact that M' is reached from M by firing t is represented by $M[t\rangle M'$. A sequence of transitions $\sigma = t_1 t_2 \ldots t_k$ is a firing sequence of $\langle \mathcal{N}, M_0 \rangle$ if there exist a sequence of markings such that $M_0[t_1\rangle M_1[t_2\rangle \cdots M_{k-1}[t_k\rangle M_k$, it can be written as $M_0[\sigma\rangle M_k$, and M_k is said to be reachable from M_0 by firing σ. A step at a marking M is a maximal set of transitions concurrently firable from M.

The reachability set $R(\mathcal{N}, M_0)$ is the set of all markings reachable from M_0. $L(\mathcal{N}, M_0)$ is the language of firing sequences of $\langle \mathcal{N}, M_0 \rangle$ ($L(\mathcal{N}, M_0) = \{\sigma | M_0[\sigma\rangle\}$).

A marked graph (MG) is a Petri net such that each place has exactly one input transition and exactly one output transition. MG's allow synchronization but no choice. MG's are a subclass of ordinary Petri nets for which a simple, powerful, and elegant theory allows very efficient analysis and synthesis algorithms. A summary of structure theory of MG's can be found in [21].

2.2 Implicit places and MG's

An *implicit place* never is the unique restricting the firing of its output transitions. Let $\mathcal{N}$ be any net and $\mathcal{N}^p$ be the net resulting from adding a place p to $\mathcal{N}$. If M_0 is an initial marking of $\mathcal{N}$, M_0^p denotes the initial marking of $\mathcal{N}^p$ and $m_0(p) = M_0^p[p]$. The incidence matrix of $\mathcal{N}$ is C and l_p is the incidence vector of place p.

DEFINITION 2.1 [22] *Let $\langle \mathcal{N}, M_0 \rangle$ be a net system and $p \notin P$ be a place to be added. Then p is an implicit place (IP) with respect to $\langle \mathcal{N}, M_0 \rangle$ (or equivalently, it is an implicit place in $\langle \mathcal{N}^p, M_0^p \rangle$) iff the languages of firing sequences of $\langle \mathcal{N}, M_0 \rangle$ and $\langle \mathcal{N}^p, M_0^p \rangle$ coincide. That is, $L(\mathcal{N}, M_0) = L(\mathcal{N}^p, M_0^p)$.*

A place is an IP depending on the initial marking, M_0. Places which can be implicit for any M_0 are said to be *structurally implicit* (SIP). Inside the class of SIP's we are interested in the so called *marking structurally implicit places* (MSIP) whose structural characterization is given in the following result.

THEOREM 2.1 [12] *Let $\mathcal{N}$ be a net and p be a place with incidence vector l_p. The place p is an MSIP in $\mathcal{N}^p$ iff there exists $Y \geq 0$ such that $Y^T \cdot C = l_p$.*

From this characterization of an MSIP, p, a method to compute an initial marking of p making it implicit with respect to $\langle \mathcal{N}, M_0 \rangle$ is presented in [12].

In the following, we characterize a special class of MSIP's with respect to strongly connected MG's called *TT-MSIP's*. These places have only one input arc and one output arc and therefore, $\mathcal{N}^p$ will be also an MG. The row of the incidence matrix corresponding to a TT-MSIP can be obtained from the summation of rows corresponding to the places in any path from the input transition to the output transition of the place. Moreover, we characterize the

minimum initial marking making these places implicit with respect to $\langle \mathcal{N}, M_0 \rangle$ and preserving its steps.

THEOREM 2.2 *Let $\mathcal{N} = (P, T, F)$ be a strongly connected MG and $p \notin P$ be a place to be added with one input transition $t_i \in T$ ($^\bullet p = \{t_i\}$) and one output transition $t_o \in T$ ($p^\bullet = \{t_o\}$). The place p is a TT-MSIP with respect to $\mathcal{N}$ and $\forall \pi \in \mathcal{P}(t_i, t_o)$, $l_p = \sum_{p_j \in \pi} l_{p_j}$.*

Proof: If $\mathcal{N}$ is a strongly connected MG, for all paths, $\pi \in \mathcal{P}(t_i, t_o)$, of the form $t_i (= t_1) p_1 t_2 \ldots t_{k-1} p_{k-1} t_k (= t_o)$: $^\bullet p_j = \{t_j\}$, $p_j^\bullet = \{t_{j+1}\}$, $j = 1, \ldots, k-1$. Therefore, the summation of the rows in the incidence matrix corresponding to the places in π, $V = \sum_{p_j \in \pi} l_{p_j}$, verifies:
(1) $V[t] = 0$, $\forall t \notin \pi$;
(2) $V[t_i] = V[t_1] = \sum_{p_j \in \pi} l_{p_j}[t_1] =$ `if` $t_i \neq t_o$ `then` $l_{p_1}[t_1] = 1$ `else` $l_{p_1}[t_1] + l_{p_{k-1}}[t_o] = 0$;
(3) $V[t_r] = \sum_{p_j \in \pi} l_{p_j}[t_r] = l_{p_{r-1}}[t_r] + l_{p_r}[t_r] = 0$, $\forall t_r \in \pi$, $r = 2 \ldots (k-1)$;
(4) $V[t_o] = V[t_k] = \sum_{p_j \in \pi} l_{p_j}[t_k] =$ `if` $t_i \neq t_o$ `then` $l_{p_{k-1}}[t_k] = -1$ `else` $l_{p_1}[t_i] + l_{p_{k-1}}[t_k] = 0$.
That is, vector V coincides with the incidence vector, l_p, of p, and according to Theorem 2.1, p is an MSIP (with $Y[p_j] =$ `if` $p_j \in \pi$ `then 1 else 0`, $\forall p_j \in P$) and also a TT-MSIP. Q.E.D.

The following result characterizes the minimum initial marking of a TT-MSIP to be implicit *preserving all steps of the net system* $\langle \mathcal{N}, M_0 \rangle$. This marking is computed from the contents of tokens of the existing paths from the input transition of p to its output transition.

THEOREM 2.3 *Let $\langle \mathcal{N}, M_0 \rangle$ be a strongly connected and live MG, and $p \notin P$ be a TT-MSIP to be added with $^\bullet p = \{t_i\}$ and $p^\bullet = \{t_o\}$. The minimum initial marking of p to be an IP in $\langle \mathcal{N}^p, M_0^p \rangle$ preserving all steps of $\langle \mathcal{N}, M_0 \rangle$ is* $m_0^{\min}(p) = \min\{\sum_{p_j \in \pi} M_0[p_j] | \pi \in \mathcal{P}(t_i, t_o)\}$.

Proof: First we prove that p is an IP with an initial marking $m_0(p) = m_0^{\min}(p)$ (i.e., $L(\mathcal{N}, M_0) = L(\mathcal{N}^p, M_0^p)$).
$L(\mathcal{N}^p, M_0^p) \subseteq L(\mathcal{N}, M_0)$. Removing place p from $\mathcal{N}^p$, we remove constraints for firing transitions. Therefore, all sequence $\sigma \in L(\mathcal{N}^p, M_0^p)$ are also firable in $\langle \mathcal{N}, M_0 \rangle$.
$L(\mathcal{N}, M_0) \subseteq L(\mathcal{N}^p, M_0^p)$. We prove this part by contradiction. Let σ be a sequence firable in $\langle \mathcal{N}, M_0 \rangle$ but not firable in $\langle \mathcal{N}^p, M_0^p \rangle$. Let σ_1 be the maximal prefix of σ firable in $\langle \mathcal{N}, M_0 \rangle$ and $\langle \mathcal{N}^p, M_0^p \rangle$: $M_0[\sigma_1\rangle M$ and $M_0^p[\sigma_1\rangle M^p$. Obviously, $M^p[p_i] = M[p_i]$ for all $p_i \in P$. The only transition preventing to finish the firing of σ after the firing of σ_1 in $\langle \mathcal{N}^p, M_0^p \rangle$ is t_o. This means that $m(p) = m_0(p) + l_p \cdot \vec{\sigma_1} = 0$. Now, we select a path $\pi \in \mathcal{P}(t_i, t_o)$ such that $m_0(p) = \sum_{p_j \in \pi} M_0[p_j]$. Moreover, according to Theorem 2.2, $l_p = \sum_{p_j \in \pi} l_{p_j}$. Therefore, substituting these last expressions in the above expression of $m(p)$ we obtain, $0 = m(p) = \sum_{p_j \in \pi} M_0[p_j] + \sum_{p_j \in \pi} l_{p_j} \cdot \vec{\sigma_1} = \sum_{p_j \in \pi} M[p_j]$. But this

contradicts the hypothesis from which σ is firable in $\langle \mathcal{N}, M_0 \rangle$ and therefore the place $p_j \in {}^\bullet t_o$ in the path π must contain at least one token.

In order to prove that $m_0(p) = m_0^{\min}(p)$ is the minimum initial marking making p an IP preserving the steps of $\langle \mathcal{N}, M_0 \rangle$ we distinguish two cases.

Case 1 ($t_i \neq t_o$, i.e., p is self-loop free). In this case, since p is an IP for $m_0(p)$, it is step preserving [11]. We prove that $m_0(p)$ is the minimum initial marking. First we build a sequence, σ, of maximal length in $\langle \mathcal{N}, M_0 \rangle$ firing only transitions of $T \setminus \{t_i\}$. All reached markings throughout the sequence are different, on the contrary we have a reproducible sequence without transition t_i and this is not possible in MG's. This sequence is finite because the number of different markings in a bounded net is finite. Since the sequence is maximal, we reach a marking M from which t_i is the unique firable transition (the net system is live), $M_0[\sigma\rangle M$.

In $\langle \mathcal{N}, M \rangle$ there exists a path, π', from t_i to t_o where all places contain zero tokens. In effect, the only firable transition from M is t_i, then all transitions of $T \setminus \{t_i\}$ have at least one input place with zero tokens. Therefore, t_o has an empty input place with an input transition that has another empty input place, and so on. This sequence cannot be a circuit because the MG is live and then one of the places in the sequence is an output place of t_i.

Taking into account that p is an IP with respect to $\langle \mathcal{N}, M_0 \rangle$, σ is also firable in $\langle \mathcal{N}^p, M_0^p \rangle$ and the number of tokens in p is: $m(p) = m_0(p) + l_p \cdot \vec{\sigma}$. Let π be a path of $\mathcal{P}(t_i, t_o)$ such that $m_0(p) = \sum_{p_j \in \pi} M_0[p_j]$. Moreover, according to Theorem 2.2, $l_p = \sum_{p_j \in \pi} l_{p_j} = \sum_{p_k \in \pi'} l_{p_k}$, because $\pi' \in \mathcal{P}(t_i, t_o)$, but in general $m_0(p) \leq \sum_{p_k \in \pi'} M_0[p_k]$. Considering these expressions, we can rewrite the contents of tokens of p in the following way: $m(p) = \sum_{p_j \in \pi} M_0[p_j] + \left(\sum_{p_j \in \pi} l_{p_j}\right) \cdot \vec{\sigma} \leq \sum_{p_k \in \pi'} M_0[p_k] + \left(\sum_{p_k \in \pi'} l_{p_k}\right) \cdot \vec{\sigma} = \sum_{p_k \in \pi'} M[p_k] = 0$.

Therefore, $m_0(p)$ is a minimal initial marking because there exists a firable sequence in $\langle \mathcal{N}^p, M_0^p \rangle$ that empties the place.

Case 2 ($t_i = t_o$, i.e., p is a self-loop). In this case, the minimal initial marking to make p an IP is equal to one. We prove that in order to preserve the steps of $\langle \mathcal{N}, M_0 \rangle$ we need at least the initial marking stated.

From $\langle \mathcal{N}, M_0 \rangle$ we can obtain a new net $\langle \mathcal{N}', M_0' \rangle$ by splitting the transition t_i into two transitions t and t' such that: ${}^\bullet t = {}^\bullet t_i$ and $t'^\bullet = t_i{}^\bullet$; and a new ordinary place p_i such that: ${}^\bullet p_i = \{t\}$ and $p_i{}^\bullet = \{t'\}$ and $M_0'[p_i] = 0$. Let $\mathcal{M}$ be the set of reachable markings of $\langle \mathcal{N}', M_0' \rangle$ in which the marking of place p_i is equal to zero. It is trivial to verify that the set $\mathcal{M}$ projected with respect to the set of places P coincides with the set of reachable markings of $\langle \mathcal{N}, M_0 \rangle$. Therefore, the set of steps of $\langle \mathcal{N}, M_0 \rangle$ is enclosed in the set of steps of $\langle \mathcal{N}', M_0' \rangle$ renaming the appearings of t by t_i and removing the appearings of t'.

Let us consider a place p with ${}^\bullet p = \{t'\}$ and $p^\bullet = \{t\}$ with respect to $\langle \mathcal{N}', M_0' \rangle$. Applying the previous Case 1 to place p we conclude that the minimum initial marking to make implicit place p with respect to $\langle \mathcal{N}', M_0' \rangle$ and preserving the steps of the net is equal to the minimal contents of tokens in the paths from t' to t (i.e., the circuits of the net system $\langle \mathcal{N}, M_0 \rangle$ traversing the transition t_i). Therefore, according to the previous paragraph a self-loop, p, with this initial

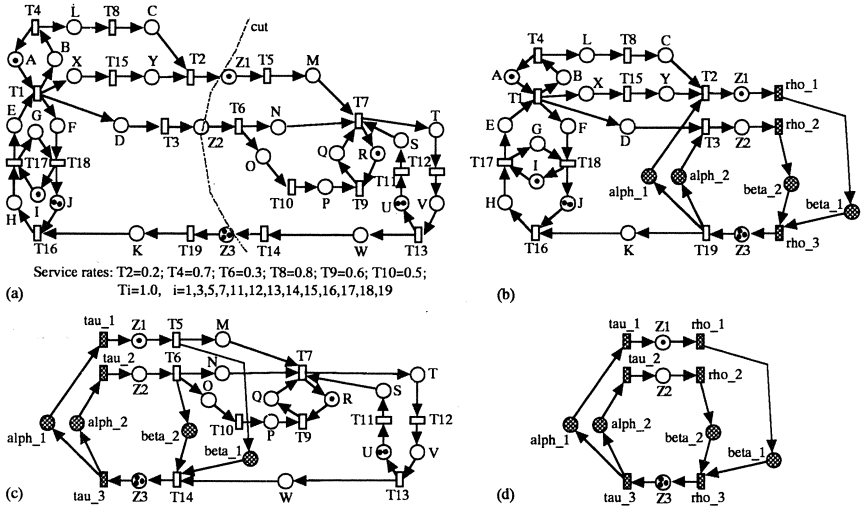


FIGURE 1. An SMG (a), its decomposition in aggregated systems $\mathcal{AS}_1$ (b), $\mathcal{AS}_2$ (c), and the basic skeleton (d).

marking preserves the steps of $\langle \mathcal{N}, M_0 \rangle$. Moreover, it is minimal because the steps requiring the maximum amount of tokens in p are the steps of $\langle \mathcal{N}, M_0 \rangle$ (they contain the output transition of p). Q.E.D.

COROLLARY 2.1 *Let $\langle \mathcal{N}, M_0 \rangle$ be a strongly connected and live MG, and $p \notin P$ be a TT-MSIP to be added with ${}^\bullet p = \{t_i\}$ and $p^\bullet = \{t_o\}$. The place p is an IP in $\langle \mathcal{N}^p, M_0^p \rangle$ preserving all steps of $\langle \mathcal{N}, M_0 \rangle$ for all initial markings $m_0(p) \geq m_0^{\min}(p)$.*

Proof: If we remove $m_0(p) - m_0^{\min}(p)$ tokens from p, then p is an IP in $\langle \mathcal{N}^p, M_0^p \rangle$ preserving all steps of $\langle \mathcal{N}, M_0 \rangle$ (Theorem 2.3). Therefore, all sequences and steps of $\langle \mathcal{N}, M_0 \rangle$ are firable in $\langle \mathcal{N}^p, M_0^p \rangle$. On the other hand, $m_0(p) - m_0^{\min}(p)$ tokens in p are frozen, hence the sequences and steps of $\langle \mathcal{N}^p, M_0^p \rangle$ coincide with those of $\langle \mathcal{N}, M_0 \rangle$. In effect, add the place p to a net system $\langle \mathcal{N}^{p'}, M_0^{p'} \rangle$ where p' is a place such that ${}^\bullet p' = \{t_i\}$, $p'^\bullet = \{t_o\}$ and its initial marking is $m_0^{\min}(p')$. $\langle \mathcal{N}^{p'}, M_0^{p'} \rangle$ has the same sequences and steps that $\langle \mathcal{N}, M_0 \rangle$ (Theorem 2.3), and there exists a sequence in $\langle \mathcal{N}^{p'}, M_0^{p'} \rangle$ that empties the place p'. The place p is identical to the place p', hence the minimum marking of p is reached when p' is empty (i.e. it contains $m_0(p) - m_0^{\min}(p)$ tokens). Q.E.D.

The Theorem 2.3 characterizes the minimum initial marking of a TT-MSIP to be an IP with respect to $\langle \mathcal{N}, M_0 \rangle$ in terms of the contents of tokens of the paths $\mathcal{P}(t_i, t_o)$. The computation of this minimum initial marking can be done applying an algorithm from the graph theory to determine the cost of the shortest path from a source vertex to a sink vertex of a directed graph,

$G = (V, E)$, obtained from the original MG (see [2] for implementations of these algorithms). In this graph, each vertex corresponds to a transition of the net. There exists a directed arc between two vertices if and only if there exists a place in the net connecting the two transitions that represent the two vertices. The sense of the arc is the sense of the tokens' flow between the transitions through the place. Each arc has a non-negative cost equal to the initial marking of the place that represents. Moreover, we add an arc $t \rightarrow t$ for each vertex t with a cost equal to ∞.

Therefore, if we apply the algorithm to solve the shortest path problem in the directed graph G, we obtain the smallest length of any path from t_i to t_o, denoted length(t_i, t_o). Observe, that length$(t_i, t_o) = \min\{\sum_{p_j \in \pi} M_0[p_j] | \pi \in \mathcal{P}(t_i, t_o)\} = m_0^{\min}(p)$.

3 Structural Decomposition of MG's

The basic idea is the following: a strongly connected and live MG (see Fig. 1.a) is split into two subnets by a *cut* Q defined through some places ($Q = \{Z1, Z2, Z3\}$, in Fig. 1.a). From the cut we define three nets: two *aggregated subnets* ($\mathcal{AN}_1$ and $\mathcal{AN}_2$; see Figs. 1.b and 1.c) and a *basic skeleton* net ($\mathcal{BN}$; see Fig. 1.d). These nets will be obtained by substitution of the so called aggregable subnets, defined from the cut Q, by a set of places. We select an initial marking for each added place such that the behaviour of the aggregated subnet is the behaviour of the original MG hiding the behaviour of the aggregable subnet.

Definition 3.1 *Let $\mathcal{N} = (P, T, F)$ be a strongly connected MG. A subset of places, $Q \subseteq P$, is said to be a cut of $\mathcal{N}$ iff there exist two subnets, $\mathcal{N}_1 = (P_1, T_1, F_1)$ and $\mathcal{N}_2 = (P_2, T_2, F_2)$, of $\mathcal{N}$ verifying*
i) $T_1 \cup T_2 = T$, $T_1 \cap T_2 = \emptyset$
ii) $P_1 = T_1^{\bullet} \cup {}^{\bullet}T_1$, $P_2 = T_2^{\bullet} \cup {}^{\bullet}T_2$
iii) $P_1 \cup P_2 = P$, $P_1 \cap P_2 = Q$
iv) $F_i = F \cap ((P_i \times T_i) \cup (T_i \times P_i))$, $i \in \{1, 2\}$

Definition 3.2 *Let $\mathcal{N} = (P, T, F)$ be a strongly connected MG, $Q \subseteq P$ a cut of $\mathcal{N}$, and $\mathcal{N}_1 = (P_1, T_1, F_1)$, $\mathcal{N}_2 = (P_2, T_2, F_2)$ the two subnets associated with the cut (by Def. 3.1). The subnets $\mathcal{N}_{A_i} = (P_{A_i}, T_{A_i}, F_{A_i})$, $i \in \{1, 2\}$, are called the aggregable subnets of the cut Q, where*
i) $P_{A_i} = P_i \setminus Q$
ii) $T_{A_i} = T_i \setminus (T_Q \cap T_i)$, *where* $T_Q = {}^{\bullet}Q \cup Q^{\bullet}$
iii) $F_{A_i} = F_i \cap ((P_{A_i} \times T_{A_i}) \cup (T_{A_i} \times P_{A_i}))$
The places $p \in P_{A_i}$ such that ${}^{\bullet}p \cap T_{A_i} = \emptyset$ (resp., $p^{\bullet} \cap T_{A_i} = \emptyset$) are called source places (resp., sink places) of $\mathcal{N}_{A_i}$. The set of input transitions of the source places and output transitions of the sink places are called interface transitions of $\mathcal{N}_{A_i}$.

We denote $\mathcal{P}_{A_i}$ the set of paths in the net $\mathcal{N}_{A_i}$ from a source place to a sink place. $\mathcal{IP}_{A_i}$ denotes the set of TT-MSIP's with respect to $\mathcal{N}$ obtained from each path of $\mathcal{P}_{A_i}$ by the linear combination of the rows in the incidence

matrix corresponding to the path's places. In the sequel, we define the so called *aggregated subnets* of an MG $\langle \mathcal{N}, M_0 \rangle$ with respect to a cut Q. These subnets will be obtained by substituting in $\mathcal{N}$ of an aggregable subnet $\mathcal{N}_{A_i}$ by the set of places $\mathcal{IP}_{A_i}$. This substitution is an abstraction of the subnet $\mathcal{N}_{A_i}$. We select an initial marking for each place $p \in \mathcal{IP}_{A_i}$ (called *aggregation's initial marking*, $m_0^a(p)$) equal to $m_0^a(p) = \min\{\sum_{p_j \in \pi} M_0[p_j] | l_p = \sum_{p_j \in \pi} l_{p_j} \text{ and } \pi \in \mathcal{P}_{A_i}\}$. With this initial marking we prove that the behaviour of the aggregated subnet is the behaviour of the original MG by hiding the behaviour of $\mathcal{N}_{A_i}$.

DEFINITION 3.3 *Let $\langle \mathcal{N}, M_0 \rangle$ be a strongly connected and live MG, $Q \subseteq P$ a cut of $\mathcal{N}$, and $\mathcal{N}_{A_i}$, $i = 1, 2$, be the aggregable subnets defined by the cut Q. The aggregated subsystem $\mathcal{AS}_i = \langle \mathcal{AN}_i, M_0^{\mathcal{AN}_i} \rangle$ is the net system obtained from $\langle \mathcal{N}, M_0 \rangle$ by substituting the subnet $\mathcal{N}_{A_j}$ by the set of places $\mathcal{IP}_{A_j}$ with $m_0(p) = m_0^a(p)$, for all $p \in \mathcal{IP}_{A_j}$, $i = 1, 2; j = 1, 2$ and $j \neq i$. The basic skeleton system, $\mathcal{BS} = \langle \mathcal{BN}, M_0^{\mathcal{BN}} \rangle$, is the system obtained from $\langle \mathcal{N}, M_0 \rangle$ by substituting the subnets $\mathcal{N}_{A_1}$ and $\mathcal{N}_{A_2}$ by the set of places $\mathcal{IP}_{A_1}$ and $\mathcal{IP}_{A_2}$ with $m_0(p) = m_0^a(p) = \min\{\sum_{p_j \in \pi} M_0[p_j] | l_p = \sum_{p_j \in \pi} l_{p_j}$ and $\pi \in \mathcal{P}_{A_i}\}$, for all $p \in \mathcal{IP}_{A_1} \cup \mathcal{IP}_{A_2}$.*

THEOREM 3.1 *Let $\langle \mathcal{N}, M_0 \rangle$ be a strongly connected and live MG, $Q \subseteq P$ a cut of $\mathcal{N}$ and $\mathcal{AS}_i$ be the aggregated subsystem obtained from $\langle \mathcal{N}, M_0 \rangle$ by substituting the subnet $\mathcal{N}_{A_j}$ by the set of places $\mathcal{IP}_{A_j}$ with $M_0^{\mathcal{AN}_i}[p] =$ `if` $p \in \mathcal{IP}_{A_j}$ `then` $m_0^a(p)$ `else` $M_0[p]$, $i = 1, 2; j = 1, 2$, and $j \neq i$.*
i) $L(\mathcal{N}, M_0)|_{T \setminus T_{A_j}} = L(\mathcal{AN}_i, M_0^{\mathcal{AN}_i})$.
ii) $R(\mathcal{N}, M_0)|_{P \setminus P_{A_j}} = R(\mathcal{AN}_i, M_0^{\mathcal{AN}_i})|_{P_{\mathcal{AN}_i} \setminus \mathcal{IP}_{A_j}}$.

Proof: $L(\mathcal{N}, M_0)|_{T \setminus T_{A_j}} \subseteq L(\mathcal{AN}_i, M_0^{\mathcal{AN}_i})$. If we add the places of $\mathcal{IP}_{A_j}$ to $\langle \mathcal{N}, M_0 \rangle$ then $L(\mathcal{N}, M_0)$ is preserved because all places of $\mathcal{IP}_{A_j}$ are IP with respect to $\langle \mathcal{N}, M_0 \rangle$ preserving its steps (Corollary 2.1, taking into account that $m_0^a(p) \geq m_0^{\min}(p)$). All sequences firable in this net are also firable in the net $\mathcal{AS}_i$ after the removing of transitions in T_{A_j}. This is because in $\mathcal{AS}_i$ we have removed all firing constraints appearing in $\langle \mathcal{N}, M_0 \rangle$ imposed by $\mathcal{N}_{A_j}$.
$L(\mathcal{AN}_i, M_0^{\mathcal{AN}_i}) \subseteq L(\mathcal{N}, M_0)|_{T \setminus T_{A_j}}$. We prove this part by contradiction. Let σ be a sequence of $L(\mathcal{AN}_i, M_0^{\mathcal{AN}_i})$ for which there is no $\sigma' \in L(\mathcal{N}, M_0)$ such that $\sigma = \sigma'|_{T \setminus T_{A_j}}$. Let σ_0 be the maximal prefix of σ for which there is a sequence $\sigma_0' \in L(\mathcal{N}, M_0)$ verifying $\sigma_0 = \sigma_0'|_{T \setminus T_{A_j}}$. If $M_0[\sigma_0'\rangle M$ and $M_0^{\mathcal{AN}_i}[\sigma_0\rangle M^{\mathcal{AN}_i}$, it is trivial to verify that $M[p] = M^{\mathcal{AN}_i}[p]$ for all $p \in (P \setminus P_{A_j})$. The next transition to σ_0, t, in σ must be an output transition of a sink place of $\mathcal{N}_{A_j}$, because these transitions are the unique transitions of $\mathcal{AS}_i$ with additional constraints to fire in $\langle \mathcal{N}, M_0 \rangle$. These constraints arise from $\mathcal{N}_{A_j}$ but not from the places $\mathcal{IP}_{A_j}$ because they are implicit with respect to $\langle \mathcal{N}, M_0 \rangle$. All maximal firable sequences in $\langle \mathcal{N}, M \rangle$ containing only transitions of $\mathcal{N}_{A_j}$ never can enable the transition t because σ_0 is the maximal prefix of σ for which there is a sequence $\sigma_0' \in L(\mathcal{N}, M_0)$ verifying $\sigma_0 = \sigma_0'|_{T \setminus T_{A_j}}$. Let M' be a marking reachable in $\langle \mathcal{N}, M \rangle$ firing a maximal sequence, σ_1, in $\langle \mathcal{N}, M \rangle$ containing only transitions

of $\mathcal{N}_{A_j}$. At M' there exists an empty path in the $\mathcal{N}_{A_j}$ from a source place to a sink place that inputs to transition t. In effect, at M' all transitions of $\mathcal{N}_{A_j}$ are not enabled, hence have at least one empty input place. Moreover, t has at least one empty input place being a sink place of $\mathcal{N}_{A_j}$ because t is not enabled at M'. Therefore, t has an empty input place whose input transition has an empty input place, and so on, until we reach a source place of $\mathcal{N}_{A_j}$. This means that a place in $\mathcal{IP}_{A_j}$ corresponding to this path is an input place of t containing zero tokens, but this contradicts the hypothesis from which t is firable in $\mathcal{AS}_i$.
$R(\mathcal{N}, M_0)|_{P \setminus P_{A_j}} = R(\mathcal{AN}_i, M_0^{\mathcal{AN}_i})|_{P_{\mathcal{AN}_i} \setminus \mathcal{IP}_{A_j}}$. To prove this, observe that $P \setminus P_{A_j} = P_{\mathcal{AN}_i} \setminus \mathcal{IP}_{A_j}$ from the definition of $\mathcal{AN}_i$. Taking into account the part (i) of this theorem, the stated equality of markings' sets holds. Q.E.D.

COROLLARY 3.1 *Let $\langle \mathcal{N}, M_0 \rangle$ be a strongly connected and live MG, $Q \subseteq P$ a cut of $\mathcal{N}$, and $\mathcal{BS}$ the basic skeleton system obtained from $\langle \mathcal{N}, M_0 \rangle$ by substituting the subnets $\mathcal{N}_{A_1}$ and $\mathcal{N}_{A_2}$ by the set of places $\mathcal{IP}_{A_1}$ and $\mathcal{IP}_{A_2}$, respectively, and $M_0^{\mathcal{BN}}[p] =$* `if` $p \in \mathcal{IP}_{A_1} \cup \mathcal{IP}_{A_2}$ `then` $m_0^a(p)$ `else` $M_0[p]$.
i) $L(\mathcal{N}, M_0)|_{T \setminus (T_{A_1} \cup T_{A_2})} = L(\mathcal{BN}, M_0^{\mathcal{BN}})$.
ii) $R(\mathcal{N}, M_0)|_{P \setminus (P_{A_1} \cup P_{A_2})} =$
$R(\mathcal{BN}, M_0^{\mathcal{BS}})|_{P_{\mathcal{BN}} \setminus (\mathcal{IP}_{A_1} \cup \mathcal{IP}_{A_2})}$.

Proof: The proof of the corollary can be decomposed into two steps: (1) The proof of the behaviour equivalence between $\langle \mathcal{N}, M_0 \rangle$ and $\mathcal{AS}_1$ (that is, the previous theorem); (2) The proof of the behaviour equivalence between $\mathcal{AS}_1$ and $\mathcal{BS}$. Taking into account that $\mathcal{AS}_i$ is a strongly connected and live MG, the proof of this second part is the same as that of the above theorem renaming, for example, $\mathcal{AS}_1$ as $\langle \mathcal{N}, M_0 \rangle$ and $\mathcal{BS}$ as $\mathcal{AS}_2$. Q.E.D.

The main drawback of the above theorems concerns the great number (exponential in the worst case) of places in $\mathcal{IP}_{A_i}$. In the following we present a method to reduce the number of places to add, characterizing a subset of $\mathcal{IP}_{A_i}$, denoted $\mathcal{BIP}_{A_i}$, with the property that all places of $\mathcal{IP}_{A_i} \setminus \mathcal{BIP}_{A_i}$ are implicit with respect to the places $\mathcal{BIP}_{A_i}$. Therefore, in order to build the aggregated subnet we only add the set of places $\mathcal{BIP}_{A_i}$ instead of $\mathcal{IP}_{A_i}$.

Let us consider the aggregable subnet $\mathcal{N}_{A_i}$ together with its interface transitions. We derive from this net a directed graph $G_{A_i} = (V, E)$ in the same way to that presented at the end of previous section.

If we apply the algorithm of R.W. Floyd to solve the *all-pairs shortest paths* problem (see [2] for implementations of this algorithm) to the directed graph G_{A_i}, we obtain for each ordered pair of vertices (i.e., transitions) (t, t') the smallest length of any path from t to t', denoted length(t, t') (if this value is equal to ∞, there is no path from t to t'). Observe, that length$(t, t') = \min\{\sum_{p_j \in \pi} M_0[p_j] | \pi$ is a path from t to $t'\}$. The computational complexity of this algorithm is $O(m^3)$, where m is the number of transitions of the considered net. From these values we define the set of places $\mathcal{BIP}_{A_i}$ as $\mathcal{BIP}_{A_i} = \{p | {}^\bullet p = \{t\}; p^\bullet = \{t'\}; t, t' \in T_Q; \text{length}(t, t') \neq \infty\}$.

For all $p \in \mathcal{BIP}_{A_i}$ we select an initial marking $m_0(p) = \text{length}(t, t')$. It is trivial to verify that this initial marking coincides with the previously defined *aggregation's initial marking*, $m_0^a(p)$. For instance, in the case of Fig. 1.b, $\mathcal{BIP}_{A_2} = \{$ *beta_1*, *beta_2* $\}$ and $m_0(beta\_1) = m_0(beta\_2) = 0$.

The following result states that all places of $\mathcal{IP}_{A_i} \setminus \mathcal{BIP}_{A_i}$ are implicit with respect to the places $\mathcal{BIP}_{A_i}$. Therefore, in order to build the aggregated subsystem we only add the set of places $\mathcal{BIP}_{A_i}$ instead of $\mathcal{IP}_{A_i}$.

PROPERTY 3.1 *Each place $p \in \mathcal{IP}_{A_i} \setminus \mathcal{BIP}_{A_i}$ with an initial marking equal to $m_0^a(p)$ is implicit with respect to the set of places $\mathcal{BIP}_{A_i}$ each one with an initial marking equal to the aggregation's marking.*

Proof: Let $p \in \mathcal{IP}_{A_i} \setminus \mathcal{BIP}_{A_i}$ be a place obtained from the summation of the rows in the incidence matrix corresponding to the places of a path. Let $t, t' \in T_Q$ be the interface transitions of this path. Because of the existence of this path, after the application of the Floyd's algorithm we have $\text{length}(t, t') \neq \infty$, therefore there exists an identical place in $\mathcal{BIP}_{A_i}$ with the same initial marking. Q.E.D.

In many cases the set $\mathcal{BIP}_{A_i}$ is bigger than necessary because some places can be implicit in $\mathcal{AS}_i$. In order to remove one of these unnecessary places, p, we can apply the method described at the end of the previous section to compute the shortest path from ${}^\bullet p$ to $p^\bullet$. The place p can be removed if the output of this algorithm is less than or equal to the aggregation's marking of p. Observe that in the case of Fig. 2.b, the set $\mathcal{BIP}_{A_2}$ contains 16 places but a further removing of places leads to a minimum set of 6 places, named β_i, $i = 1, \ldots, 6$ in the figure.

4 ITERATIVE TECHNIQUE FOR APPROXIMATE THROUGHPUT COMPUTATION

In the previous section, an algorithm to decompose an MG into two aggregated subsystems and a basic skeleton system (being also MG's) has been presented. In aggregated subsystem $\mathcal{AS}_i$ $(i = 1, 2)$, the subnet $\mathcal{N}_j$ $(j \neq i)$ is represented by the places in the cut Q, by the interface transitions of $\mathcal{N}_j$, $T_{I_j} = T_Q \cap T_j$, and by the new places that substitute the subnet $\mathcal{N}_{A_j}$.

The technique for an approximate computation of the throughput that we present now is, basically, a *response time approximation* method [1, 16, 17]. The interface transitions of $\mathcal{N}_j$ in $\mathcal{AS}_i$ approximate the response time of all the subsystem $\mathcal{N}_j$ $(i = 1, 2; j \neq i)$. A direct (non-iterative) method to compute the constant service rates of such interface transitions in order to represent the aggregation of the subnet gives, in general, low accuracy. Therefore, we are forced to define a *fixed-point search iterative process*, with the possible drawback of the presence of convergence and efficiency problems.

4.1 First approach: Ping-Pong algorithm

The first algorithm that we explored, called "Ping-Pong", follows.

select a cut Q;
derive aggregated subsystems $\mathcal{AS}_i, i = 1, 2$;
give value μ_t^0 for each $t \in T_{I_1}$ in $\mathcal{AS}_2$;
compute value of throughput χ_2^0 of $\mathcal{AS}_2$;
$k := 0$; {counter for iteration steps}
repeat
  $k := k + 1$;
  compute μ_t^k for each $t \in T_{I_2}$ such that the throughput
    χ_1^k of $\mathcal{AS}_1$ is close enough to χ_2^{k-1};
  compute μ_t^k for each $t \in T_{I_1}$ such that the throughput
    χ_2^k of $\mathcal{AS}_2$ is close enough to χ_1^k;
until convergence of χ_1^k and χ_2^k;

In the above procedure, once a cut has been selected and given some initial values for the service rates of interface transitions of $\mathcal{N}_1$ (which approximate the response time of all the subsystem $\mathcal{N}_1$), the underlying CTMC of aggregated subsystem $\mathcal{AS}_2$ is solved. From the solution of that CTMC, the first estimation χ_2^0 of the throughput of $\mathcal{AS}_2$ can be computed. Then, the initial estimated values of service rates of interface transitions that approximate the response time of subsystem $\mathcal{N}_2$ must be derived. This must be done in such a way that the throughput χ_1^1 of $\mathcal{AS}_1$ is "close enough" to χ_2^0. Then, a better estimation of rates μ_t^k for each $t \in T_{I_1}$ must be computed such that the throughput χ_2^1 of $\mathcal{AS}_2$ is close enough to χ_1^1. The process is iterated until χ_i^{k-1} and χ_i^k are "close enough".

The first problem of the above sketch of approximation algorithm is that a *multidimensional search on the parameters* of a complex CTMC in order to get a given throughput cannot be done in an efficient way. A possible solution to this problem is the following. In the iterative process, each time that an aggregated subsystem $\mathcal{AS}_i, i = 1, 2$, is solved, *the ratios* among the service rates μ_t^k of all the transitions in T_{I_i} are estimated. After that, when the other subsystem $\mathcal{AS}_j, j \neq i$, is solved, only a *scale factor* for these service rates must be computed. The goal is to find a scale factor of μ_t^k for all $t \in T_{I_j}$ (and fixed k) such that the throughput of $\mathcal{AS}_j$ and the throughput of $\mathcal{AS}_i$, computed before, are the same. And this can be achieved with a linear search of the scale factor in $\mathcal{AS}_j$.

At this point, the main technical problem is the following: How to estimate from the solution of $\mathcal{AS}_i$ the ratios among the service rates of all transitions in T_{I_i} that in the next step (solution of $\mathcal{AS}_j$) will be scaled to obtain an approximation of the response time of the subsystem $\mathcal{N}_i$?

We explain our answer to this question by means of the example depicted in Fig. 1. Figure 1.b represents the aggregated subsystem $\mathcal{AS}_1$ derived from the original MG. It is necessary to compute the ratio between the service rate of $T2$ and $T3$ to be used as input data for the linear search of the scale factor in $\mathcal{AS}_2$ (Fig. 1.c). In order to do that, the aggregated subsystem $\mathcal{AS}_1$ is transformed (as depicted in Fig. 1.b) with the addition of places $\mathcal{BIP}_{A_1} = \{alph\_1, alph\_2\}$. The obtained system is behaviourally equivalent to $\mathcal{AS}_1$ because the added places (which are those that will substitute $\mathcal{N}_{A_1}$), are implicit. These new

$\mathcal{AS}_1$					$\mathcal{AS}_2$				
χ_1	tau_1	tau_2	tau_3	coeff	χ_2	rho_1	rho_2	rho_3	coeff
0.17352	0.05170	0.16810	0.88873	1.01167	0.12714	0.89026	0.21861	0.14354	0.98468
0.14093	0.06265	0.19707	0.91895	1.01318	0.13795	0.88267	0.21363	0.13509	0.98582
0.13856	0.06325	0.19821	0.92054	1.01306	0.13841	0.88239	0.21343	0.13467	0.98592
0.13844	0.06328	0.19827	0.92062	1.01306	0.13843	0.88237	0.21342	0.13465	0.98592
0.13843	0.06328	0.19827	0.92064	1.01307	0.13843	0.88238	0.21342	0.13465	0.98593

TABLE 1. Iteration results for the SMG in Fig. 1.

places allow to estimate the ratio between the "aggregated service times" of transitions $T2$ and $T3$ (representing the response time approximation of $\mathcal{N}_1$), as the quotient of the mean marking of *alph_1* by the mean marking of *alph_2*, because the throughput of all transitions is the same.

Now, two problems arise. First, the linear search of the scale factor must be done in the aggregated subsystems, that can have a considerably large state space, thus the efficiency of the method falls down. Additionally, we have found convergence problems in many cases. A solution for both problems is proposed in the next subsection.

4.2 A solution: Pelota[1] *algorithm*

The more practical solution of the problem we found makes use of the third system (another MG) derived from the original one, in the previous section: *the basic skeleton.* The basic skeleton contains the interface subsystem and a simplified view (using the places $\mathcal{BIP}_{A_i}, i = 1, 2$, computed by the algorithm in previous section) of subsystems $\mathcal{N}_{A_i}, i = 1, 2$.

The idea is to use the basic skeleton as an intermediate point (*fronton*) between the two aggregated subnets (rackets), as explained in this algorithm:

select a cut Q;
derive $\mathcal{AS}_i, i = 1, 2$ and $\mathcal{BS}$;
give initial value μ_t^0 for each $t \in T_{I_2}$;
$k := 0$; {counter for iteration steps}
repeat
  $k := k + 1$;
  solve aggregated subsystem $\mathcal{AS}_1$ with
    input: μ_t^{k-1} for each $t \in T_{I_2}$,
    output: ratios among μ_t^k of $t \in T_{I_1}$ and χ_1^k;
  solve basic skeleton system $\mathcal{BS}$ with
    input: μ_t^{k-1} for each $t \in T_{I_2}$,
      ratios among μ_t^k of $t \in T_{I_1}$, and χ_1^k,
    output: scale factor of μ_t^k of $t \in T_{I_1}$;
  solve aggregated subsystem $\mathcal{AS}_2$ with
    input: μ_t^k for each $t \in T_{I_1}$,
    output: ratios among μ_t^k of $t \in T_{I_2}$ and χ_2^k;
  solve basic skeleton system $\mathcal{BS}$ with

[1] Game played by two players who use a basket strapped to their wrists or a wooden racket to propel a ball against a specially marked wall, called *fronton.*

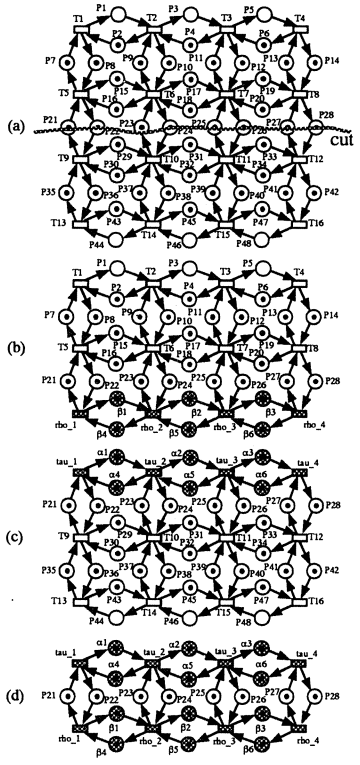


FIGURE 2. A second example of SMG and its decomposition.

input: μ_t^k for each $t \in T_{I_1}$,
ratios among μ_t^k of $t \in T_{I_2}$, and χ_2^k,
output: scale factor of μ_t^k of $t \in T_{I_2}$;
until convergence of χ_1^k and χ_2^k;

In this iterative process, each time that an aggregated subsystem $\mathcal{AS}_i, i = 1, 2$, is solved, only the throughput χ_i^k and the ratios among the service rates μ_t^k of all the transitions in T_{I_i} are estimated (with the method explained in the previous subsection). After that, a scale factor for these service rates must be computed. This is achieved by using the basic skeleton system $\mathcal{BS}$. The goal is to find a scale factor of μ_t^k for all $t \in T_{I_i}$ such that the throughput of the basic skeleton and the throughput of $\mathcal{AS}_i$, computed before, are the same. A linear search of the scale factor must be implemented, but now in a net system with considerably fewer states (the basic skeleton). In each iteration of this linear search, the basic skeleton is solved by deriving the underlying CTMC.

Now, the existence and uniqueness of the solution, and the convergence of the method should be addressed. Although no formal proof gives positive

answers so far to the above questions, extensive testing allows the conjecture that there exists one and only one solution, computable in a finite number of steps, typically between 2 and 5 if the convergence criterion is that the difference between the two last estimations of the throughput is less than 0.1 %.

5 EXAMPLES

In this section we present several numerical results of the application of the iterative technique previously introduced. Among all the tested examples, we have selected two different Petri net structures because of their following characteristics: the first one (already introduced in Fig. 1) is structurally asymmetric while the second has symmetries; for the second one, the effect of timing asymmetries on the iterative algorithm can be studied by changing the service rates of transitions (preserving the strong structural symmetry). In all cases, the obtained approximations are compared with exact values obtained from the numerical solution of the underlying CTMC (*GreatSPN* package was used [9]).

Let us consider again the SMG depicted in Fig. 1.a. The exact value of the throughput is equal to 0.138341 (if single-server semantics is assumed). The underlying CTMC has 89358 states. The aggregated systems $\mathcal{AS}_1$ and $\mathcal{AS}_2$ are depicted in Figs. 1.b and 1.c, respectively. The corresponding basic skeleton system is that in Fig. 1.d.

Table 1 shows the iterative results obtained for this example. The values in $\mathcal{AS}_1$ columns have been obtained from the solution of the aggregated system in Fig. 1.b: χ_1 is the throughput of $\mathcal{AS}_1$; columns *tau_1*, *tau_2*, and *tau_3* are the estimated values of the service rates of the aggregated transition tau_1, tau_2, and tau_3, computed in $\mathcal{AS}_1$; column *coeff* is the scale factor of previous estimated service rates, obtained by the linear search in the basic skeleton of Fig. 1.d. Columns related with $\mathcal{AS}_2$ represent the analogous values for the aggregated system in Fig. 1.c. Convergence of the method can be observed from the third iteration step. The error is -0.064333 %, after the fifth step. The following additional fact must be remarked: the underlying CTMC's of $\mathcal{AS}_1$, $\mathcal{AS}_2$, and the basic skeleton have 8288, 3440, and 231 states, respectively, while the original SMG has 89358 states.

As a second example, let us consider the SMG depicted in Fig. 2.a. Any splitting of the net will generate two strongly coupled aggregated subnets. We select the following cut: $Q = \{P21, P22, P23, P24, P25, P26, P27, P28\}$. The corresponding aggregated systems are depicted in Figs. 2.b and 2.c. The basic skeleton is that in Fig. 2.d. The CTMC underlying the original SMG has 49398, while those underlying the aggregated systems have 6748. The basic skeleton has 771 reachable states.

We consider three different situations arising from different transition service rates (we assume infinite-server semantics in all cases). In the first case, we suppose that the service rates of all transitions are equal to 1.0. In this case, the exact throughput of the SMG is 0.295945.

Table 2 shows the iteration results for three different selections of initial values of aggregated service rates of transitions *rho_1*, *rho_2*, and *rho_3*. It can be seen that in all cases convergence occurs at the third iteration step,

Initial values of service rates of rho_1, rho_2, and rho_3 equal to 0.1											
AS_1						AS_2					
x_1	tau_1	tau_2	tau_3	tau_4	coeff	x_2	rho_1	rho_2	rho_3	rho_4	coeff
0.07930	1.02121	1.02452	1.01112	0.80930	1.06357	0.33294	0.29834	0.50973	0.61599	0.71668	1.03611
0.29244	0.84574	0.72462	0.55755	0.30802	1.05833	0.30079	0.29864	0.54035	0.70609	0.83610	1.06250
0.29710	0.84301	0.71383	0.54364	0.29813	1.06382	0.29733	0.29758	0.54270	0.71310	0.84299	1.06427
0.29711	0.84340	0.71354	0.54286	0.29751	1.06436	0.29711	0.29747	0.54281	0.71352	0.84343	1.06440

Initial values of service rates of rho_1, rho_2, and rho_3 equal to 1.0											
AS_1						AS_2					
x_1	tau_1	tau_2	tau_3	tau_4	coeff	x_2	rho_1	rho_2	rho_3	rho_4	coeff
0.33318	0.70982	0.61546	0.51044	0.29917	1.03518	0.29265	0.30871	0.55771	0.72423	0.84521	1.05804
0.30095	0.83571	0.70581	0.54034	0.29877	1.06233	0.29712	0.29817	0.54366	0.71378	0.84293	1.06378
0.29734	0.84296	0.71307	0.54270	0.29759	1.06425	0.29712	0.29751	0.54286	0.71354	0.84339	1.06436
0.29712	0.84343	0.71352	0.54281	0.29747	1.06440	0.29710	0.29746	0.54282	0.71354	0.84345	1.06439

Initial values of service rates of rho_1, rho_2, and rho_3 equal to 10.0											
AS_1						AS_2					
x_1	tau_1	tau_2	tau_3	tau_4	coeff	x_2	rho_1	rho_2	rho_3	rho_4	coeff
0.33419	0.68611	0.59756	0.49474	0.28053	1.03311	0.28561	0.30812	0.56325	0.73687	0.85741	1.06091
0.30136	0.83550	0.70455	0.53890	0.29791	1.06293	0.29679	0.29807	0.54392	0.71447	0.84356	1.06392
0.29735	0.84299	0.71304	0.54263	0.29753	1.06430	0.29710	0.29750	0.54287	0.71358	0.84343	1.06437
0.29711	0.84343	0.71352	0.54281	0.29747	1.06440	0.29710	0.29746	0.54282	0.71355	0.84346	1.06441
0.29710	0.84346	0.71355	0.54282	0.29746	1.06441	0.29710	0.29745	0.54281	0.71355	0.84346	1.06441
0.29710	0.84346	0.71356	0.54282	0.29746	1.06441	0.29710	0.29746	0.54282	0.71355	0.84346	1.06441

TABLE 2. Iteration results for the SMG in Fig. 2 with all service rates of transition equal to 1.0.

AS_1						AS_2					
x_1	tau_1	tau_2	tau_3	tau_4	coeff	x_2	rho_1	rho_2	rho_3	rho_4	coeff
0.33318	0.70983	0.61546	0.51045	0.29917	1.03519	0.34424	0.70118	1.49390	1.84123	1.92737	1.06187
0.33352	0.71500	0.60522	0.49835	0.28554	1.03637	0.33345	0.68342	1.50320	1.85362	1.93598	1.06255
0.33345	0.71616	0.60538	0.49834	0.28550	1.03656	0.33345	0.68281	1.50288	1.85352	1.93592	1.06251
0.33345	0.71621	0.60539	0.49834	0.28550	1.03656	0.33345	0.68278	1.50284	1.85348	1.93588	1.06249

TABLE 3. Iteration results for the SMG in Fig. 2 with service rates of transition $T1$ to $T8$ equal to 1.0 and of transitions $T9$ to $T16$ equal to 2.0.

independently of the initial values given to the aggregated service rates. This fact illustrates the robustness of the method with respect to the seed. The error of the approximation in all cases is 0.4%.

As a second case, consider again the SMG in Fig. 2.a but now with asymmetric service rates associated with transitions. Assume that the service rates of transitions $T1$ to $T8$ are all equal to 1.0, while service rates of transition $T9$ to $T16$ are equal to 2.0. In this case the exact throughput of the original system is 0.333356. The iteration results are shown in Table 3. Now, the initial values of aggregated service rates of transitions *rho_1*, *rho_2*, and *rho_3* are equal to 1.0. Convergence can be observed from the second iteration step and the error of the obtained value is 0.02%. Finally, consider once more the SMG of Fig. 2.a, but now with the following service rates associated with transitions: the rates of $T1$, $T2$, $T5$, $T6$, $T9$, $T10$, $T13$, and $T14$ are equal to 2.0, while the rest are equal to 1.0. In this case, the exact throughput is 0.362586. The iteration results are shown in Table 4. Again, convergence can be observed from the second iteration step. The error is now 0.19%.

AS_1						AS_2					
x_1	tau_1	tau_2	tau_3	tau_4	coeff	x_2	rho_1	rho_2	rho_3	rho_4	coeff
0.40526	1.64486	1.58029	0.60759	0.36042	1.00568	0.35214	0.36948	0.69530	0.61363	0.80667	1.07872
0.36392	1.81297	1.72253	0.66348	0.38291	1.01927	0.36239	0.37446	0.68764	0.59809	0.79673	1.08484
0.36326	1.80988	1.72268	0.66584	0.38570	1.01711	0.36321	0.37514	0.68748	0.59702	0.79565	1.08508
0.36328	1.80942	1.72245	0.66596	0.38596	1.01688	0.36328	0.37520	0.68748	0.59694	0.79556	1.08510
0.36329	1.80938	1.72243	0.66598	0.38599	1.01686	0.36329	0.37521	0.68747	0.59693	0.79555	1.08510
0.36329	1.80938	1.72243	0.66598	0.38599	1.01686	0.36329	0.37521	0.68748	0.59694	0.79555	1.08510

TABLE 4. Iteration results for the SMG in Fig. 2 with service rates of transition $T1$, $T2$, $T5$, $T6$, $T9$, $T10$, $T13$, and $T14$ equal to 2.0, and the rest equal to 1.0.

6 CONCLUSIONS

In order to derive a general, efficient, and accurate technique for throughput approximation of stochastic marked graphs using the divide and conquer principle, qualitative theory ought to guide the decomposition phase (this principle underlies several previous works on the topic [6, 14, 16, 17, 18, 19]).

This was the first objective of the paper: the presentation of a general structural decomposition technique allowing to split a given marked graph through an arbitrary cut (subset of places) and to derive two aggregated subsystems whose qualitative behaviours are projections of the whole system qualitative behaviour. The technical tool to achieve this problem has been the use of implicit places: a subsystem of the original marked graph can be substituted by a minimal set of implicit places that represent an abstraction of the subsystem, leading to an aggregated subsystem. If the same process is applied to two complementary subsystems, two aggregated subsystems are derived, each one representing a portion of the behaviour of the whole system.

The second phase of the analysis problem is the selection of an approximate throughput computation algorithm. Iterative response time approximation technique was selected after a wide comparison with other approaches present in the literature. In order to assure the convergence of the method, a third subsystem was used for a correct tuning of parameters, the basic skeleton. It is obtained after the substitution of both subnets by the corresponding implicit places. Its behaviour is simple enough to allow a linear search of the correct value of a parameter in order to get a given throughput (the one obtained in the previous iteration step).

Extensive numerical experiments using the method sketched in previous paragraphs showed very good results with respect to efficiency and accuracy. Convergence is generally observed after a couple of iteration steps and the approximate computation of throughput can be achieved with a considerable saving of time and memory (more than one order of magnitude) and with a very small error (less than 3%).

Even though we have considered only strongly connected nets, the approach can be applied to non-strongly connected marked graphs: The iterative technique is used to compute the approximate throughput of each strongly connected component in isolation and, after that, [8, Theorem 5.1] applies.

An obvious generalization of the presented technique can be derived if the original system is partitioned into more than two subsystems, leading to the

classical tradeoff between efficiency and accuracy. The extension to more general net subclasses, like macroplace-macrotransition nets proposed in [14], is being considered by the authors.

REFERENCES

1. S. C. Agrawal, J. P. Buzen, and A. W. Shum. Response time preservation: A general technique for developing approximate algorithms for queueing networks. In *Proc. of the 1984 ACM Sigmetrics Conf. on Measurement and Modeling of Computer Systems*, pp. 63–77, Cambridge, MA, Aug. 1984.
2. A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *Data Structures and Algorithms.* Addison-Wesley, 1983.
3. M. Ajmone Marsan, G. Balbo, and G. Conte. *Performance Models of Multiprocessor Systems.* MIT Press, Cambridge, 1986.
4. H. H. Ammar and S. M. R. Islam. Time scale decomposition of a class of generalized stochastic Petri net models. *IEEE Trans. Software Eng.*, 15(6):809–820, June 1989.
5. F. Baccelli and Z. Liu. Comparison properties of stochastic decision free Petri nets. *IEEE Trans. Automat. Contr.*, 37(12):1905–1920, Dec. 1992.
6. B. Baynat and Y. Dallery. Approximate techniques for general closed queueing networks with subnetworks having population constraints. Tech. report, MASI 90-49, Univ. Paris 6, Paris, France, Oct. 1990.
7. B. Baynat and Y. Dallery. A unified view of product-form approximation techniques for general closed queueing networks. Tech. report, MASI 90-48, Univ. Paris 6, Paris, France, Oct. 1990.
8. J. Campos, G. Chiola, J. M. Colom, and M. Silva. Properties and performance bounds for timed marked graphs. *IEEE Trans. Circuits and Syst.—I: Fundamental Theory and Applications*, 39(5):386–401, May 1992.
9. G. Chiola. A graphical Petri net tool for performance analysis. In *Proc. of the 3rd Int. Workshop on Modeling Techniques and Performance Evaluation*, Paris, France, March 1987. AFCET.
10. G. Ciardo and K. Trivedi. A decomposition approach for stochastic Petri nets models. In *Proc. of the 4th Int. Workshop on Petri Nets and Performance Models*, pp. 74–83, Melbourne, Australia, Dec. 1991. IEEE Comput. Soc. Press.
11. J. M. Colom. *Análisis Estructural de Redes de Petri, Programación Lineal y Geometría Convexa.* PhD thesis, Dpto. de Ingeniería Eléctrica e Informática, Univ. Zaragoza, Spain, June 1989.
12. J. M. Colom and M. Silva. Improving the linearly based characterization of P/T nets. In G. Rozenberg, editor, *Advances in Petri Nets 1990*, Vol. 483 of *LNCS*, pp. 113–145. Springer-Verlag, Berlin, 1991.
13. Y. Dallery, Z. Liu, and D. Towsley. Equivalence, reversibility and symmetry properties in fork/join queueing networks with blocking. Tech. report, MASI 90-32, Univ. Paris 6, Paris, France, June 1990.
14. A. Desrochers, H. Jungnitz, and M. Silva. An approximation method for the performance analysis of manufacturing systems based on GSPN's. In *Proc.*

of the Rensselaer's Third Int. Conf. on Computer Integrated Manufacturing, pp. 46–55, Troy, NY, May 1992. IEEE Comput. Soc. Press.

15. S. Donatelli and M. Sereno. On the product form solution for stochastic Petri nets. In *Proc. of the 13th Int. Conf. on Applications and Theory of Petri Nets*, pp. 154–172, Sheffield, UK, June 1992.
16. H. Jungnitz, B. Sánchez, and M. Silva. Approximate throughput computation of stochastic marked graphs. *Journal of Parallel and Distributed Computing*, 15:282–295, 1992.
17. H. J. Jungnitz. *Approximation Methods for Stochastic Petri Nets.* PhD thesis, Dept. of Electrical, Computer and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY, May 1992.
18. Y. Li and C. M. Woodside. Iterative decomposition and aggregation of stochastic marked graphs Petri nets. In *Proc. of the 12th Int. Conf. on Applications and Theory of Petri Nets*, pp. 257–275, Gjern, Denmark, June 1991.
19. Y. Li and C. M. Woodside. Performance Petri net analysis of communications protocol software by delay-equivalent aggregation. In *Proc. of the 4th Int. Workshop on Petri Nets and Performance Models*, pp. 64–73, Melbourne, Australia, Dec. 1991. IEEE Comput. Soc. Press.
20. R. A. Marie. An approximate analytical method for general queueing networks. *IEEE Trans. Software Eng.*, 5(5):530–538, Sep. 1979.
21. T. Murata. Petri nets: Properties, analysis, and applications. *Proceedings of the IEEE*, 77(4):541–580, April 1989.
22. M. Silva. *Las Redes de Petri en la Automática y la Informática.* Editorial AC, Madrid, 1985.

Marking Optimization and Parallelism of Marked Graphs

Miguel Canales, Bruno Gaujal*
INRIA Centre Sophia-Antipolis
B.P. 93, 06902 Sophia-Antipolis Cedex, France

The aim of the paper is to provide a formalization of the notion of parallelism of a marked graph exploitable by parallel simulation. We show that there exists an optimal starting point for equational simulations which gives a speed of simulation in the order of the intrinsic sequentiality of the system. Furthermore, under few assumptions, the modification of the marking will accelerate the simulation without altering its results for the stationary regime. We also derive algorithms to compute this optimal marking.

1 INTRODUCTION

Marked graphs constitute a good formalism to model manufacturing systems combining parallel tasks and synchronizations. They have been extensively studied either in the deterministic or in the stochastic context [9],[3]. These systems were shown to have a linear behavior if considered in the semi-field (max,+) and this property was the starting point of an extensive ergodic theory developed by F. Baccelli [3] and of a new method of parallel simulation based on recursive equations introduced in [9] and used in [4] and [13]. These parallel simulations of marked graphs are of a new kind: they are not really event driven as in [14] or [18], but rather, equation "driven". The evolution is described by the successive application of linear transformations to the state variables. The aim of this paper is to improve the efficiency of these simulations by changing its starting state which is the initial marking of the graph.

Marking optimizations have been obtained in the deterministic case [20]. Here, we introduce the marking M^* which allows one to run parallel equational simulations more efficiently. Indeed, the cost of these algorithms depends on $L(M)$, the longest path without tokens in the marked graph under the initial marking M. The marking M^* will be chosen so that $L(M^*)$ is minimum. We show that $L(M^*)$ roughly equals the intrinsic sequentiality of the system so that little hope of increasing the speed of these algorithms is left. We also provide an algorithm to compute the couple $(M^*, L(M^*))$. Then, we show that changing the starting point of the system (i.e. its initial marking) will not

*Supported by the European Grant BRA-QMIPS of CEC DG XIII.

alter the stationary behavior of the system, provided that the system is stable of course, under fairly general assumptions.

In section 2, we give some preliminaries, in section 3, we give a formal meaning to the notion of intrinsic parallelism of a marked graph. In section 4, we define the marking M^* and we derive a computation of this marking. In the fifth section, we show that the evolution of a stochastic marked graph that satisfies the conditions of stability does not depend on the initial marking. This result is true for strongly connected graphs and for open systems with a single initial component. In the last section we apply the notions introduced in sections 3, 4 and 5 to give the optimal starting point for a parallel simulation of a marked graph.

2 PRELIMINARIES

In the preliminaries, we will describe the model of marked graph that we will use in the following. This model is more precisely presented in [2].

A *marked graph* is a Petri Net where each place has exactly one input transition and one output transition. We will denote it by $E = (P, T, A)$ where T is the set of transitions, P the set of places and A is the set of the links. A is included in $P \times T \cup T \times P$. We denote by ${}^\bullet p$ the transition preceding place p and by ${}^\bullet t$ the set of the input places of transition t. $p^\bullet$ denotes the output transition of p and $t^\bullet$ is the set of the output places of transition t. The vector M denotes the marking of the net; $M(p)$ represents the number of tokens in place p. We will denote the graph along with the marking by $G = (E, M) = (P, T, A, M)$.

Now we introduce the temporized model. We attached durations to the firing of transitions and to the holding time of places.

$S = (E, \Phi, \Sigma, Y, U, M_0)$ is a timed marked graph if (E, M_0) is a marked graph, $\Phi = (\phi_t(n))_{t,n}$ is the set of the firing time sequences of the transitions, $\Sigma = (\sigma_p(n))_{p,n}$ is the set of the holding time sequences in the places, $Y = (Y(p,l))_{p,l}$ is the set of the lag times of the initial tokens and $U = (u_i(n))_{i,n}$ is the set of the arriving sequences of the inputs.

- $\phi_t(n)$ denotes the duration of the n-th firing time of transition t. If transition t begins to fire for the n-th time at epoch e, this firing will end at time $e + \phi_t(n)$ and at this very moment, tokens are taken out of the places in ${}^\bullet t$ and put in the places in $t^\bullet$.

- $\sigma_p(n)$ is the holding time in place p of the n-th token to enter this place. If the n-th token enters place p at epoch e, it is not available for enabling the transition in $p^\bullet$ before epoch $e + \sigma_p(n)$.

- $Y(p,l)$ is defined only for $l \leq M_0(p)$. It represents the lag time of the l-th initial token in place p. The lag time of a token is a holding time of an initial token (it does not come from the firing of a transition).

- If the system has inputs (i.e. transitions with no input places) the firing of these transitions is determined by the sequences $(u_i(n))$. The input transition i fires for the n-th time at epoch $u_i(1) + \cdots + u_i(n)$.

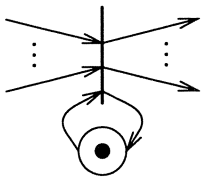


FIGURE 1. A token in the place recycling each transition guarantees the FIFO discipline.

We make some basic assumptions on our model.

FIFO Assumption:

First, all the transitions and all the places are assumed to be FIFO.

For every transition t, the n-th firing completion (token departure) of t corresponds to the n-th firing that t has started. A simple condition for a transition to be FIFO is to be *recycled* (i.e. it is the output and the input transition of a place with one token in the initial marking, as is shown in Figure 1). In the following, the transitions are always recycled. The place recycling transition t is denoted p_t.

For an arbitrary place p, the FIFO condition says that the n-th token to enter that place must become available for firing of the output transitions before the $(n+1)$-st token to enter this place. In other words tokens cannot overtake each other in the places. A simple condition for places to be FIFO is to consider only constant holding times in the places: $\forall n, \sigma_p(n) = \sigma_p$. In the following this assumption will always be fulfilled.

Initial Conditions:

- Once the initial marking M_0 is given, a non-timed net is defined by (E, M_0). We denote by $R(M)$ the set of all the markings reachable from M. It is well known that $M \in R(M')$ is a parallelism relation if E is a marked graph [24].

- The lag times must verify two conditions to be weakly compatible ([2] p. 70): The lag times must respect the FIFO feature. This means that for any place p, $Y(p,1) \leq \cdots \leq Y(p, M(p))$. Moreover, lag times of tokens in place p cannot exceed the firing time of the transition ${}^\bullet p$ plus the holding time in place p. This means that this token could be the result of a firing of ${}^\bullet p$. In the stochastic case, the lag time must belong to the support of the distributions of the firing times of ${}^\bullet p$ plus the holding time of p. Note that the lag times in the places recycling the transitions (there is only one token in these places) are not constrained by the first condition of weak compatibility. This remark will be useful in the following.

Stochastic Assumptions:

All the random variables considered here are defined on a common probability space (Ω, F, P).

The holding time sequences as well as the firing time sequences are sequences of non-negative real random vectors.

We make the following assumptions (see [3] for further insight on these assumptions):

- *Stationarity and ergodicity:*

  The sequences $\{\sigma_t(n)\}$ and $\{\phi_p(n)\}$ are ergodic and stationary for all t and p.

- *Integrability:*

  The random variables $\sigma_t(n)$ and $\phi_p(n)$ are integrable.

- *Coupled ergodicity:*

  If the system has several inputs with temporizations $(u_1(n), \cdots, u_k(n))$, all the variables $(u_i(n))$, $(u_i(n) - u_j(k))$ are jointly ergodic and stationary for all couples of inputs i, j and for all n, k.

We denote by $M(S, e)$ the marking in the system S at epoch e and by $M(p, e)$ the marking in place p at epoch e.

3 Parallelism in a Marked Graph

Measures of parallelism for discrete event systems are presented in [5], [15]. However, we are looking for a parallelism exploitable by parallel equation driven simulations and we present notions useful in this particular frame. Since the efficiency of parallel simulations depends on the availability of sufficient parallelism in the model itself, we present the notion of sequentiality of a marked graph. It will be further related to the complexity of the simulation algorithm we are interested in. In other words, we will see that the sequentiality is an appropriate measure of the parallelism present in the marked graph.

3.1 Graph of Precedence

In this section we consider a marked graph without the timings: $G = (E, M_0)$.

We introduce a different structure based on the dependence relations in the marked graph. Such a graph belongs to the class of task graphs called PERT (or activity network [12]). This graph is also called the developed graph and is presented in [7] for example. However our view is slightly different since we are not interested in analyzing the performance of the system but in discovering the available parallelism present in it.

The notion of parallelism present in a marked graph will be independent of the temporizations of the transitions and only be based on the precedence relations between the firing epochs.

In that purpose, we construct a graph of precedence τ which can be constructed in the following way. The graph τ has $T \times I\!N \cup \{\perp\}$ as the set of

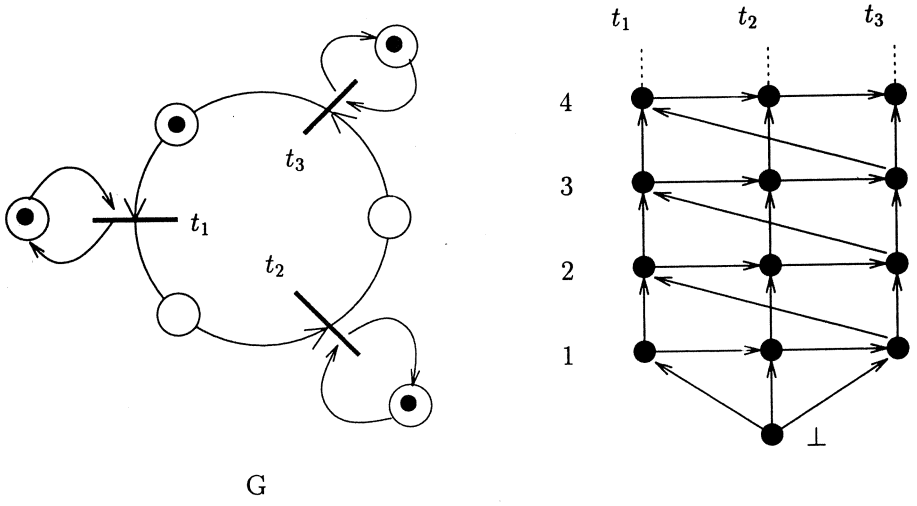


FIGURE 2. A marked graph and its associated graph of precedence. Since all the transitions are recycled, each column of τ is a total order.

vertices. The vertex $\perp$ is special and represents the "initiation" of the graph of precedence. τ has an edge between the vertex (t, n) and the vertex (t', n') if the n'-th firing of transition t' uses a token produced by the n-th firing of transition t. We also put the edges $((\perp), (t, 1)), \cdots, ((\perp), (t, m(t)))$ for all transitions t, where $m(t) = \max_{p \in \pi(t)}(M_0(p))$. Figure 2 depicts a marked graph and its graph of precedence.

We establish a few properties that help to see the relations between the two structures. We do not pretend to be exhaustive here. Indeed there exist a lot of relations between the graph of precedence and the marked graph. We will present only the few properties useful in the following.

Remark: *The graph τ does not essentially depend on M_0 but rather on the class $R(M_0)$.*

Let M_1 be a marking in $R(M_0)$ and τ_1 its associated graph of precedence.

We show that τ and τ_1 are nearly isomorphic. Consider the *firing vector* $(n_1, \cdots, n_{|T|})$, n_i being the number of times transition t_i has to be fired to go from marking M_0 to marking M_1. If there is an arc $((t_1, k_1), (t_2, k_2))$ in τ, this means that there is a place between transitions t_1 and t_2 with $k_2 - k_1$ tokens under marking M_0. Now, we know that the number of tokens in this place under M_1 is $(k_2 - k_1) + (n_1 - n_2)$. Therefore there is an arc in τ_1 between $(t_1, k_1 - n_1)$ and $(t_2, k_2 - n_2)$ as long as $k_1 - n_1 > 0$. If $k_1 - n_1 \leq 0$ then this means that there is an arc between $\perp$ and $(t_2, k_2 - n_2)$ in τ_1.

We can construct the mapping :

$$
\begin{array}{lccll}
f: & \tau & \rightarrow & \tau_1 & \\
 & (t_i, n) & \mapsto & (t_i, n - n_i) & if \quad n > n_i \\
 & (t_i, n) & \mapsto & (\bot) & if \quad n \leq n_i \\
 & (\bot) & \mapsto & (\bot) &
\end{array}
$$

f is a morphism and nearly an isomorphism between τ and τ_1 because only a finite number of vertices are contracted into $\bot$.

Eventually, in the same way that (E, M_0) and (E, M_1) are considered equivalent, we can also consider the graphs τ and τ_1 to be equivalent.

This remark reinforces the fact that τ is a good model to study the parallelism of the marked graph. Indeed, the parallelism present in a marked graph should not depend on the starting point of the system but only on the overall "dependencies" between the events taking place during the evolution of the system.

PROPOSITION 3.1 *The marked graph G is live if and only if τ is acyclic.*

Proof:

Suppose that τ contains a cycle $((t_1, n_1), \cdots, (t_k, n_k))$. This implies that $(t_1, \cdots, t_k)$ is a cycle in the Marked Graph G. By definition of τ, this means that the n_1-th firing of transition t_1 depends on the n_k-th firing of t_k which depends on the n_{k-1}-th firing of t_{k-1} and so on. Eventually, the n_1-th firing of transition t_1 depends on the n_1-th firing of transition t_1. This implies that the cycle $(t_1, \cdots, t_k)$ does not contain any token which means that the marked Graph is not live.

Conversely, if G is not live, then it contains a cycle with no token, $t_1, \cdots, t_k$. In the graph of precedence we can exhibit the cycle $(t_1, 1), \cdots, (t_k, 1)$. ■

Remark: In the following, the marked graph will be live and τ may be considered as a partial order as a consequence. We will refer to antichains (pairwize uncomparable subsets of τ) and chains (totally ordered subsets), following the usual notations and definitions (presented in [11]).

We define the level n in τ as the set $\{(t, n), t \in T\}$. A chain up to level n in τ is an oriented path from $\bot$ to any element of the level n.

3.2 Degree of Parallelism

The *degree of parallelism* of a system is usually defined by the number of events than can take place in parallel in the system.

DEFINITION 1 *The degree of parallelism δ of a marked graph G is the length of the maximum antichain in τ.*

The degree of parallelism is equal to the maximum concurrency of a marked graph, i.e. the maximal number of transitions which are concurrently enabled at a marking M reachable from M_0. Indeed, suppose that transitions $\{t_1, \cdots, t_k\}$ form a maximum concurrent set and are enabled by marking M, which is reached from M_0 with the firing vector $(n_1, \cdots, n_{|T|})$. Then, the nodes $(t_1, n_1 +$

$1), \cdots, (t_k, n_k + 1)$ form an antichain in the graph of precedence. Conversely, if $(t_1, n_1 + 1), \cdots, (t_k, n_k + 1)$ is a maximum antichain in τ, $\{t_1, \cdots, t_k\}$ form a concurrent set and are enabled by the marking reached from M_0 with the firing vector $(n_1, \cdots, n_{|T|})$.

An algorithm to compute δ has been derived in [21] (see also [24]). Anyway, some trivial bounds on δ are easy to find. $|T|$ is an upper bound of δ since $(t, n) <_\tau (t, k)$ whenever $0 < n < k$. This will happen to be good enough for parallel simulation of marked graphs on massive parallel machines as the Connection Machine. Indeed, for most systems, it is possible to allocate at least one processor per transition.

3.3 Sequentiality

We introduce a dual notion of the degree of parallelism which will be of interest in the following.

We consider the longest sequence of totally ordered firings during an evolution of the marked graph up to the level n in τ. The longer this sequence, the less parallel the marked graph. To get a finite value, we take the ratio of this sequence over n, the level in τ. To get rid of the influence of the initial part of the graph τ that depends on the initial marking, while it does not essentially depend on it (see proposition 3.1), we take the limit to infinity.

DEFINITION 2 *The sequentiality $s(G)$ of a marked graph G is defined by:*

$$s(G) = \lim_{n \to \infty} s_n,$$

where $s_n = \dfrac{\textit{length of the longest chain in } \tau \textit{ up to level } n}{n}$.

We will show in the following that this limit exists. However, it is easy to see that $s_n \leq |T|$ for all n, so that s is bounded.

We define the critical cycle C_r in the marked graph (G, M_0) as the cycle with the maximum average length, where the average length of a cycle C is its length $l(C)$ divided by the number of tokens it contains $w(C)$. We call λ the average length of the critical cycle in (G, M_0).

$$\lambda = \max_{C \in \mathcal{C}} \frac{l(C)}{w(C)}.$$

PROPOSITION 3.2 *The average length of the critical cycle of a marked graph is equal to the sequentiality of the associated graph of task*

$$s(G) = \lambda.$$

Proof:

Similar results have been obtained earlier in [25] for example. We restate the proof in our formalism without pretending originality. Let us add temporisations in the transitions of the marked graph G . Each firing of each transition is assumed to last for a time unit. Therefore the length of the longest chain in

τ starting with $\perp$ and ending with the node (t, n) equals the time it takes in the marked graph to reach the nth firing of transition t, denoted $X_t(n)$.

But now, if one uses the language of marked graphs developed in [2] , one can use the periodicity result shown in [9] and write for any transition t, $X_t(n) = k\lambda + X_t(n-k), n > n_0$ for some bounded k and for n_0 large enough, in the case G is strongly connected. In the general case, one can also write $\max_t X_t(n) = k\lambda + \max_t X_t(n-k), \quad n > n_0$.

In terms of the graph τ, this last equality can be rewritten

$$s_n(\tau) = \frac{n_1 s_{n_1}(\tau) + \lambda(n - n_1)}{n}, \tag{1}$$

where n_1 is bounded and defined by : $n_1 = n - k\lfloor (n - n_0)/k \rfloor$. Finally when n goes to infinity in the equality (1), we get $s(\tau) = \lambda$. ■

PROPOSITION 3.3 *If $M' \in R(M)$, $G = (E, M)$ and $G' = (E, M')$, then $s(G) = s(G')$.*

Proof:

In marked graphs, the number of tokens remains constant in all the cycles (see [24]). Therefore, the critical cycles are the same in both G and G'. Since the sequentiality of a marked graph equals the average length of a critical cycle (proposition 3.2), we get the equality, $s(G') = \lambda = s(G)$. ■

3.4 PRAM Model

In this section, we consider a parallel algorithm whose task graph is τ and that would run on a PRAM machine. This sort of models is studied in [8] for example, based on notions first introduced in [19].

The typical method to do so is to allocate one processor per "column" of τ (corresponding to one transition in G). Since the degree of parallelism δ of the system is smaller than $|T|$, this allocation is optimal.

Now the time it takes to compute the firing epochs up to level n is proportional to the longest path in the graph of tasks, that is proportional to $n.s(G)$.

In the following we will compare the complexity of real implementation of the simulation of the marked graph with this theoretical performance.

4 SHORTEST LONGEST PATH WITHOUT TOKENS

We introduce yet another notion: the longest path without tokens. Let $M \in R(M_0)$. $L(M)$ is the length of the longest path in the marked graph (E, M) with no tokens. $L(M)$ is finite because (E, M) is live.

DEFINITION 3

$$L^* = \min_{M \in R(M_0)} (L(M))$$

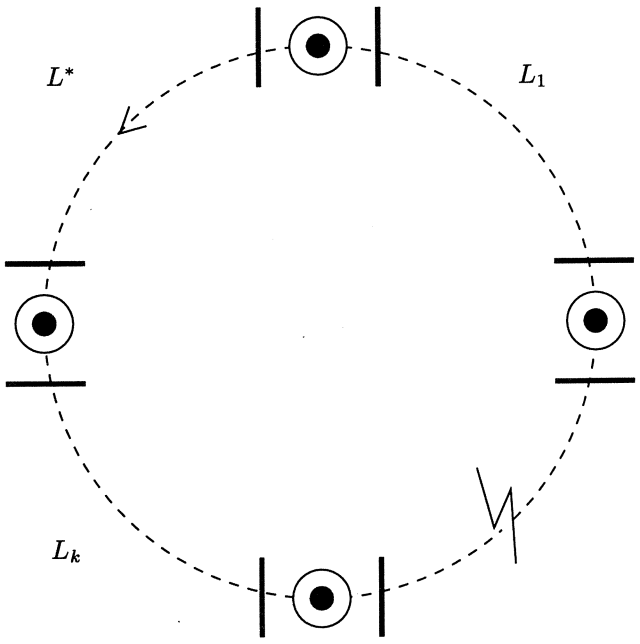


FIGURE 3. This is the form of a critical cycle in the graph under the marking M^*. For all i, $L_i = L^*$ or $L^* - 1$.

---

and M^* is a marking such that $L(M^*) = L^*$.

Note that a marking M achieving the equality $L(M) = L^*$ is not unique. However we will denote by M^* any marking verifying this equality.

PROPOSITION 4.1 $L^* = \lceil s(G) \rceil - 1$

Remark: This proposition establishes the relation between the sequentiality of the marked graph and the longest path without tokens under the appropriate marking.

Proof:

We will prove that $L^* = \lceil \lambda \rceil - 1$ which is equivalent to the result using proposition 3.2. The proof of this proposition will come from a way to compute L^*. We will show that L^* is obtained in a critical cycle which will be of the form shown in figure 3.

Let M^* be the marking in $R(M_0)$ achieving L^* as rarely as possible. Consider a path P_0 of length L^*. We construct a set S of transitions in the following way: The transition at the beginning of path P_0 (denoted t_0) belongs to S. Note that t_0 is enabled. Now consider the input places of t_0 with only one token and all the paths of length L^* or $L^* - 1$ ending in one of these places. Call these paths $P_1, P_2, \cdots P_{k_1}$. The origin transitions of these paths $t_1, t_2, \cdots, t_{k_1}$ are put in S. These transitions are all enabled. For each path P_i, we consider again all

the paths of length L^* or $L^* - 1$ ending in transition t_i. The origin transitions of all these paths are also put in S. We continue this operation until no new transition is put in S.

Suppose that no transition ending the original path P_0 with only one token in the place between this transition and P_0 belongs to S. We fire all transitions in S once. All the paths $P_1, \cdots, P_{|S|}$ are modified into paths $P'_1, \cdots, P'_{|S|}$ with the same lengths; however, path P_0 is transformed into a path P'_0 of length $L^* - 1$ so we have reduced the number of paths of length L^*. This contradicts the definition of M^*, therefore, there must exist a transition t'_0 ending the original path and belonging to S. This means that path P_0 belongs to a circuit formed by paths $P_0, P_{n_1}, P_{n_2}, \cdots, P_{n_c}$ of length L^* or $L^* - 1$.

This cycle has the form depicted in figure 3 and is called a *critical* cycle.

Now, this cycle C has say k tokens and has a length greater than $k + (k - 1)(L^* - 1) + L^*$. So $\lambda \geq 1 + L^* - 1 + 1/k = L^* + 1/k$. On the other hand, assume that $\lambda > L^* + 1 + \alpha$ with $\alpha > 0$. The critical cycle has say m tokens so its length is $m + mL^* + m\alpha$. As $\alpha > 0$ and $m\alpha$ is an integer, then, $m\alpha \geq 1$. But now, this means that one path without token in the critical cycle is longer than L^* which is impossible.

Finally, we have shown that $L^* + 1/k \leq \lambda \leq L^* + 1$. This implies that $L^* = \lceil \lambda \rceil - 1 = \lceil s(G) \rceil - 1$. ∎

Note that proposition 4.1 says that L^* is the integer approximation of the sequentiality of the system. This remark gives an insight on the reason why the complexity of the parallel simulations of a marked graph are linear in L^* in the best case. See section 6 for a detailed discussion on this topic.

4.1 Computation of (M^*, L^*)

The previous proposition of L^* allows one to derive an algorithm to compute a couple (M^*, L^*).

4.1.1 Computation of L^*

One can use Karp's theorem to compute L^*. Karp's theorem [17] provides a method to compute the maximal average weight W of a circuit in a weighted graph with n vertices. The graph is described by the weighted incidence matrix: $A_{i,j}$ = the weight of the arc (i, j). If there is no arc between i and j, $A_{i,j} = -\infty$.

Pick a vertex j.

$$W = \max_{i=1,\cdots,n} \min_{k=1,\cdots,n} \frac{(A^n)_{i,j} - (A^k)_{i,j}}{n - k}.$$

For our problem, consider the graph formed by the transitions of a marked graph. We put a weight $-M_0(i, j)$ (- the initial marking in the place between i and j) on the arc between transitions i and j. We apply Karp's algorithm to this graph and we get the cycle with the maximal average weight of a cycle, W, which corresponds to the maximal length per token of a circuit: $\lambda = -1/W$. Proposition 4.1 implies $L^* = \lceil -1/W \rceil - 1$.

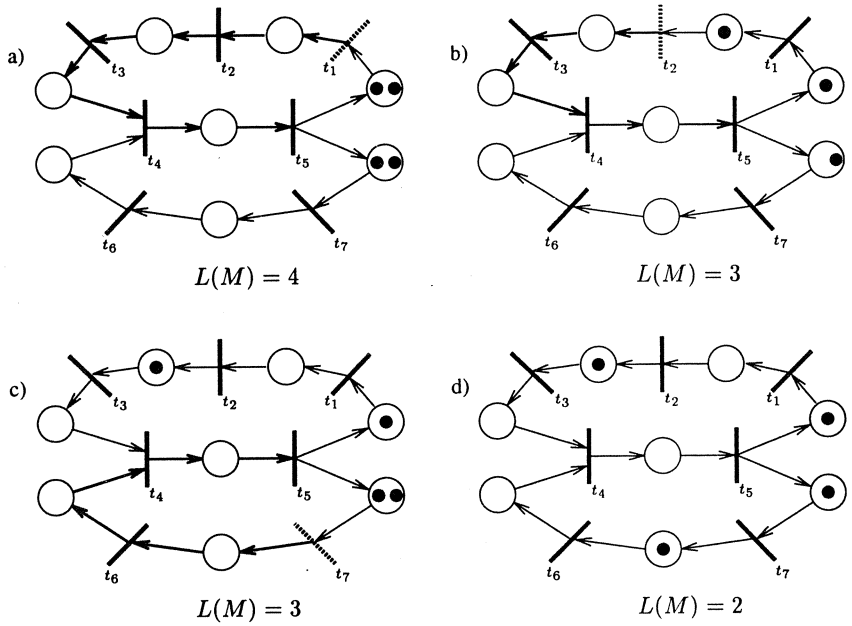


FIGURE 4. First notice that the critical cycle is of length 5 with 2 tokens. Therefore $L^* = 2$. At each step we show the path P that was picked by the algorithm with bold arcs and the set S_P as dashed transitions.

4.1.2 Computation of M^*

The previous computation does not provide a marking M^*. We derive an algorithm to compute M^*.

```
While L(M) > L* do
   choose one empty path P of length L(M).
   compute S_P'.
   fire all transitions in S_P.
```

This algorithm computes a marking of the kind M^*. Indeed, suppose M is a reachable marking but not a marking of the kind M^*. Then, using the properties of the set S_P given in the proof of proposition 4.1, the firing of all transitions in S_P for some empty path P of length $L(M)$, Will either reduce $L(M)$ or the number of paths of length $L(M)$. After some iteration $L(M)$ will decrease and eventually a marking M with $L(M) = L^*$ will be reached.

Figure 4 shows an application of this algorithm to a small marked graph.

A detailed and parallelized version of this algorithm is presented in [6]. This version does not compute L^* first but combines the computation of both

(M^*, L^*): we fire all the transitions which belong to all the sets S_P for all empty paths P of length $L(M)$. We stop when neither $L(M)$ nor the number of paths of length $L(M)$ are reduced. This version has the advantage to provide a marking M^* that minimizes the number of paths of length L^*.

5 Insensitivity With Respect To The Initial Marking

In this section, we show that for systems with a stationary regime that does not depend on the initial lag times, the initial marking does not alter this stationary regime.

For technical reasons, we distinguish two cases: strongly connected nets and open nets. However the results will be very similar in both cases.

5.1 Strongly Connected Nets

A marked graph E is strongly connected if there exists an oriented path from any transition of T to any other transition in T. In the following we will denote strongly connected marked graphs by SCMG. Note that the systems under study here have no inputs.

The following theorem has been shown in [10].

THEOREM 5.1 *Let $G_1 = (E, M_1)$ and $G_2 = (E, M_2)$ be two live SCMG which differ only in their initial markings. If $M_2 \in R(M_1)$, then by blocking an arbitrary transition t, G_1 and G_2 will reach the same marking M where no transition but t is enabled.*

Now we consider a temporized system $S = (E, \Phi, \Sigma, Y, M_0)$.

In [2] the following theorem has been established.

THEOREM 5.2 *If one transition has a firing sequence with an unbounded support distribution, the system is stable and admits a unique stationary regime regardless of the initial condition Y.*

This condition of stability can be considered fairly general. However it is not necessary. A necessary and sufficient condition of stability of SCMG is given in [22]. In fact only few and classified systems admit several stationary regimes depending on the initial lag-times (see [23]). In the following we will only use the condition given in theorem 5.2.

Now, we can formulate the following key theorem:

THEOREM 5.3 *Let $S_1 = (E, \Phi^1, \Sigma^1, Y_1, M_1)$ and $S_2 = (E, \Phi^2, \Sigma^2, Y_2, M_2)$ be two SCMG with weakly compatible lag-times and the same joint distribution of the sequence of the firing times. Assume that the firing times form jointly stationary and ergodic sequences of integrable r.v.'s and that the sequences of firing times at different servers are mutually independent. Assume that one transition has an unbounded support firing distribution. Then, conditions of stability are satisfied. If $M_2 \in R(M_1)$ then the stationary regimes of the two systems are identical.*

Proof:

Consider the systems $S_1' = (E, \Phi^1, \Sigma^1, Y_1', M_1)$ and $S_2' = (E, \Phi^2, \Sigma^2, Y_2', M_2)$ where $Y_1'(p_{t_0}, 1) = Y_2'(p_{t_0}, 1) = \infty$ for a transition t_0 with unbounded firing distribution support and $Y_1'(p,l) = Y^1(p,l), Y_2'(p,l) = Y^2(p,l) \ \ \forall p \neq p_{t_0}$.

According to theorem 5.1, these two systems will eventually reach the same marking D if t_0 is blocked. We denote by k_t^1 and k_t^2 the numbers of times transition t has fired in S_1' and S_2' respectively before reaching the marking D.

We define T_0^1 and T_0^2 by:

$$T_0^1 = \sum_{t \in T} \sum_{l=1}^{k_t^1} \phi_j^1(l) + \sum_{p \in P} k_{\bullet p}^1 \sigma_p^1 + \sum_{p \in P} \sum_{l=1}^{M_1(p)} Y_1(p,l),$$

$$T_0^2 = \sum_{t \in T} \sum_{l=1}^{k_t^2} \phi_j^2(l) + \sum_{p \in P} k_{\bullet p}^1 \sigma_p^2 + \sum_{p \in P} \sum_{l=1}^{M_2(p)} Y_2(p,l).$$

T_0^1 (resp. T_0^2) is chosen large enough so that at time T_0^1 (resp. T_0^2) the system S_1' (resp. S_2') is blocked. Transition t_0 is the only transition that is enabled.

Finally we set $T_0 = \max\{T_0^1, T_0^2\}$. At time T_0 both systems have reached the marking D.

We consider the systems $S_1'' = (E, \Phi^1, \Sigma^1, Y_1'', M_1)$ and $S_2'' = (E, \Phi^2, \Sigma^2, Y_2'', M_2)$ where $Y_1''(p_{t_0}, 1) = T_0$, $Y_2''(p_{t_0}, 1) = T_0$ and $Y_1''(p,l) = Y^1(p,l) \ \ \forall p \neq p_{t_0}$, $Y_2''(p,l) = Y^2(p,l) \ \ \forall p \neq p_{t_0}$. We obtain $m(S_1'', T_0) = m(S_2'', T_0) = D$. Note that systems S_1'' and S_2'' have weakly compatible lag-times. Furthermore, theorem 5.2 allows one to say that S_1'' and S_2'' are stable and that they have the same stationary regime as S_1 and S_2 respectively.

We just have to show that S_1'' and S_2'' have the same stationary regime.

Since the sequences of firing times are mutually independent and stationary, we can couple the firing times in S_1'' and S_2'' in the following way:

$$\phi_t^1(n + k_t^1) = \phi_t^2(n + k_t^2) \ \ \forall t \in T, \ \forall n \geq 0.$$

These sequences are independent and stationary. Under such coupling, one sees that at any epoch $e \geq T_0$, the marking in both systems is the same:

$$m(S_1'', e) = m(S_2'', e) \ \forall e \geq T_0.$$

This implies that the two original systems S_1 and S_2 have the same stationary regime. ■

We have run some experiments to give an idea of the speed of coupling. Indeed, the theorem 5.3 does not say anything on that feature.

Figure 5 depicts a Petri net consisting in a single circuit. Figure 6 shows the evolution of the average number of tokens in the place p_6, for two different simulations. In one, six tokens were assigned to place p_1 and 0 to the others, and in a second simulation the six tokens were assigned to the place p_6. The firing times of all the transitions are i.i.d. exponential variables with the same

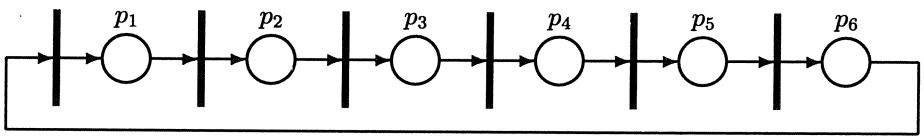


FIGURE 5. A circuit marked graph.

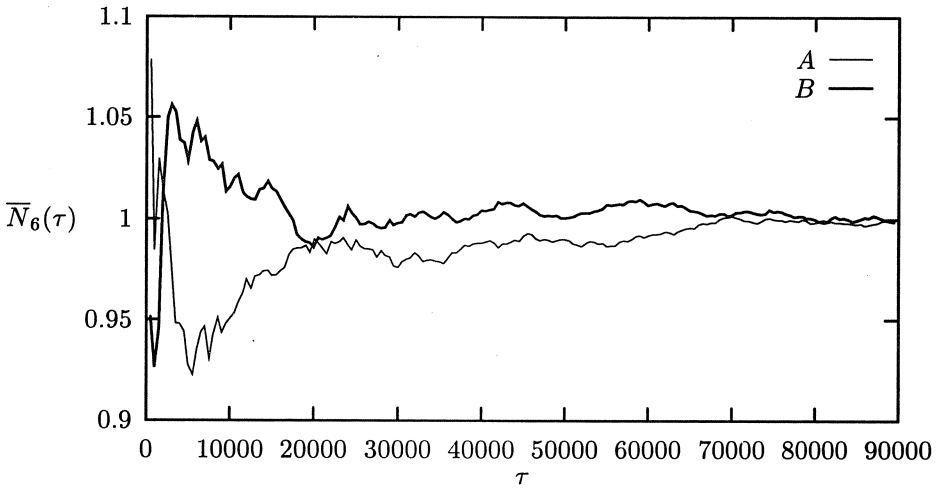


FIGURE 6. Stationary distribution of tokens for the graph depicted in Figure 5. A is the evolution of the marking in p_6 with $M_0 = (0,0,0,0,0,6)$ and B with $M_0 = (6,0,0,0,0,0)$.

parameter. One can observe in figure 6 that the convergence to the same marking does not occur before the coupling with the stationary regime for both systems as it is suggested by the proof. The speed of convergence to the stationary regime is a difficult problem [1], and it is not addressed here.

5.2 Non Strongly Connected Nets

If a marked graph is not strongly connected, it is decomposable into strongly connected components interlinked by an acyclic oriented net (I). Therefore, the components can be partially ordered by I. Let C and C' be two components. $C <_I C'$ means that there are arcs from C to C' (and no arcs from C' to C). In figure 8, the order of the components is $C_0 <_I C_1 <_I C_2$.

In particular, we call *initial components* the minimal components according to I. These initial components can be either SCMG or input transitions.

5.3 Networks With a Single Initial Component

We do not distinguish in this subsection between the case where this component is merely an input transition or an entire SCMG. We give the theorems corresponding respectively to theorems 5.1, 5.2, 5.3 in the case of a marked graph with a single initial component.

First we extend theorem 5.1 to open networks with a single input component.

THEOREM 5.4 *Let $G = (E, M)$ and $G' = (E, M')$ be two connected and live marked graphs and C_0 be the initial component of E. If one blocks any transition $t_0 \in C_0$, then if $M \in R(M')$, G and G' reach the same marking M_{t_0} where no transition can fire but t_0.*

Proof:

Since the graph E is connected, for any transition $t (t \neq t_0)$ in E, there is an oriented path from t_0 to t. Let us consider the path from t_0 to t that contains the smallest number of tokens under the current marking M. This path is simple because, since G is live, all the cycles contain tokens. However, it may not be unique. In this case, we choose the path with the smallest indices of transitions and we denote this path W_t. We shall denote by M_t the number of tokens on such a path under the marking M.

Now, block transition t_0. It is easy to see that transition t cannot fire more than M_t times before blocking. Indeed when a transition distinct from t_0 or t is fired no token is added or removed in W_t. If t is fired, tokens are removed from all the simple paths from t_0 to t and in particular from W_t. This implies that after some firings, the whole network will eventually block (no transition is enabled except t_0).

Now, we will see that t blocks after exactly M_t firings, which means that the path W_t is empty when t blocks. Let us say that the network reaches a complete deadlock under the marking D. Under D, no transition is enabled except t_0. Under D, let us follow the longest path without tokens L_t which ends in transition t. This path begins in transition t_0 otherwise this beginning transition would be enabled. Now this path from t_0 to t is empty and by the token cycle conservation law in marked graphs, this path contained M_t tokens under the initial marking so the path W_t is also empty, see figure 7.

We construct a subgraph S of E in the following way: For each transition t, $t \neq t_0$ in G we only keep one path from t_0 to t, the path W_t.

First, let us remark that S is a spanning tree of G: indeed,

- S is connected: Each transition has a least one path which links it with t_0 in E.
- S contains no cycle: The existence of a cycle in S would mean that some transition in G is linked to t_0 by more than one path.

Now, we can note that S does not depend on M. If we construct S' starting with another marking $M' \in R(M)$, we would get the same spanning tree, i.e. $S = S'$.

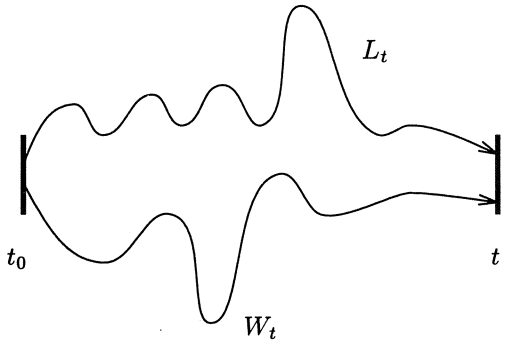


FIGURE 7. The two paths L_t and W_t are necessarily empty under marking D because L_t is empty by definition and $D(L_t) - D(W_t) = M(L_t) - M(W_t) \geq 0$.

Let t be a transition in E. Let Σ_1 and Σ_2 be 2 paths from t_0 to t. $M'(\Sigma_1) - M'(\Sigma_2) = M(\Sigma_1) - M(\Sigma_2)$. So the minimum path W_t is the same in both markings.

The final step is to describe the marking D (which will necessarily be the same for any starting marking). All the places in S are empty. Now, we add one place of E to the spanning tree S. We create a cycle. If this cycle is a circuit C then all the weight of the circuit $M(C)$ must be put in this place. Otherwise, if the edge is (t_1, t_2) consider the paths from t_0 to t_1 and t_2 respectively W_{t_1} and W_{t_2}. In the original marking, we know by construction of S that $M(W_{t_1}) + M(t_1, t_2) \geq M(W_{t_2})$. So there must be a non-negative weight $M(W_{t_1}) + M(t_1, t_2) - M(W_{t_2})$ on the edge (t_1, t_2). This marking is the same for any starting marking in $R(M)$ indeed. ∎

Now, we give the conditions of stability of an open system. These conditions are established in [2]. Here we give only the conditions of stability of a system with a single initial component.

THEOREM 5.5 *If the initial component verifies the condition of stability when considered in isolation as a strongly connected system given in theorem 5.2, and if for any components C_i and C_j, $C_i <_I C_j$ implies that the cycle time of C_i (isolated) is bigger than the cycle time of C_j (isolated), then the marked graph is stable and its unique stationary regime does not depend on the initial lag-times.*

These two results allow us to derive the insensitivity of the stationary regime for open networks with one initial component. The formulation of the theorem is similar to theorem 5.3.

THEOREM 5.6 *Let $S_1 = (E, \Phi^1, \Sigma^1, Y_1, M_1)$ and $S_2 = (E, \Phi^2, \Sigma^2, Y_2, M_2)$ be two MG with one input component and with the same joint distribution of the*

sequence of the firing and holding times. Assume that the firing and holding times form jointly stationary and ergodic sequences of integrable r.v.'s and that the sequences of firing times at different servers are mutually independent. Assume also that the system satisfies the conditions of stability and that one transition in the input component has an unbounded firing distribution. If $M_2 \in R(M_1)$ *then the stationary regimes of the two systems are identical.*

Proof: The proof is the same as for theorem 5.3.

Consider the systems $S_1' = (E, \Phi^1, \Sigma^1, Y_1', M_1)$ and $S_2' = (E, \Phi^2, \Sigma^2, Y_2', M_2)$ where $Y_1'(p_{t_0}, 1) = Y_2'(p_{t_0}, 1) = \infty$ for some transition t_0 with an unbounded support for its firing distributions and $Y_1'(p,l) = Y^1(p,l), Y_2'(p,l) = Y^2(p,l)\ \forall p \neq p_{t_0}$.

According to theorem 5.4, these two systems will reach the same marking M. We denote by k_t^1 and k_t^2 the numbers of times transition t has fired in S_1' and S_2' respectively before reaching the marking M.

We define T_0^1 and T_0^2 by:

$$T_0^1 = \sum_{t\in T}\sum_{l=1}^{k_t^1} \phi_t^1(l) + \sum_{p\in P} k_{\bullet p}^1 \sigma_p^1 + \sum_{p\in P}\sum_{l=1}^{M_1(p)} Y_1(p,l),$$

$$T_0^2 = \sum_{t\in T}\sum_{l=1}^{k_t^2} \phi_t^2(l) + \sum_{p\in P} k_{\bullet p}^2 \sigma_p^2 + \sum_{p\in P}\sum_{l=1}^{M_2(p)} Y_2(p,l),$$

and finally $T_0 = \max\{T_0^1, T_0^2\}$.

We consider the systems $S_1'' = (E, \Phi^1, \Sigma^1, Y_1'', M_1)$ and $S_2'' = (E, \Phi^2, \Sigma^2, Y_2'', M_2)$ where $Y_1''(p_{t_0}, 1) = T_0$, $Y_2''(p_{t_0}, 1) = T_0$ and $Y_1''(p,l) = Y^1(p,l)\ \ \forall p \neq p_{t_0}$, $Y_2''(p,l) = Y^2(p,l)\ \ \forall p \neq p_{t_0}$. These two systems have weakly compatible lag-times. T_0 is chosen large enough so that we obtain $m(S_1'', T_0) = m(S_2'', T_0) = M$. Furthermore, theorem 5.5 allows one to say that S_1'' and S_2'' are stable and that they have the same stationary regime as S_1 and S_2 respectively.

We just have to show that S_1'' and S_2'' have the same stationary regime.

Since the sequences of firing times are mutually independent and stationary, we can couple the firing times in S_1'' and S_2'' in the following way:

$$\phi_j^1(n + k_j^1) = \phi_j^2(n + k_j^2)\ \ \forall j \in T,\ \forall n \geq 0.$$

These sequences are also mutually independent and stationary. Under such coupling, one sees that

$$m(S_1'', t) = m(S_2'', t)\ \forall t \geq t_0.$$

Therefore, the two systems reach the same stationary regime. ■

The convergence is illustrated by an example depicted in figure 8. Figure 9 shows the evolution of the average marking in the place between transitions 4 and 5 with two different initial markings. For A we used the initial marking $(0,0,0,0,2,0,3,0)$. For B we used the initial marking $(1,1,1,1,1,1,1,0)$ which is reachable from the previous one.

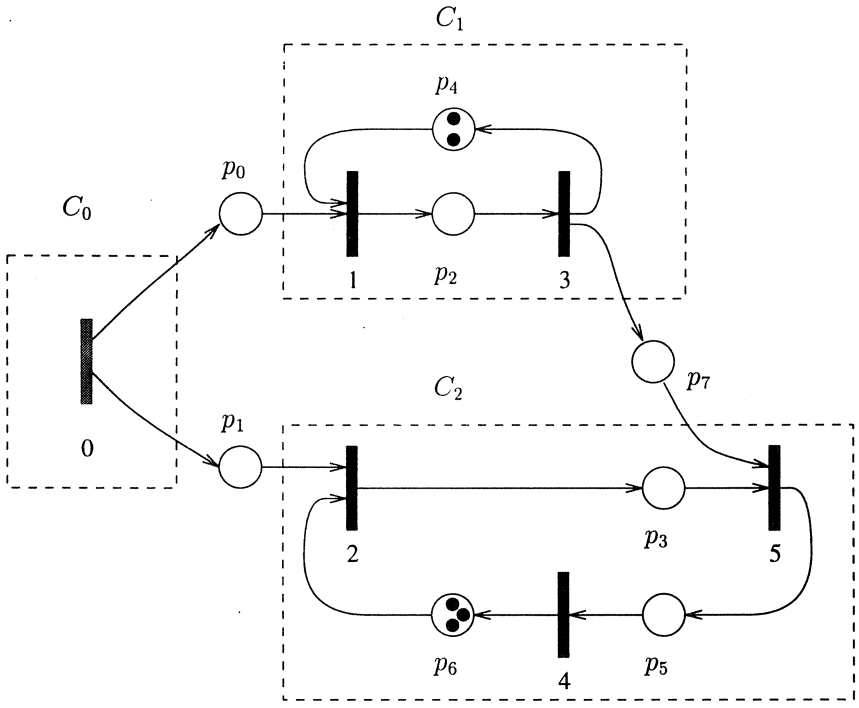


FIGURE 8. Open system with three components $C_0 < C_1 < C_2$. All the transitions are recycled with places containing one token but this is not shown in the figure for simplicity.

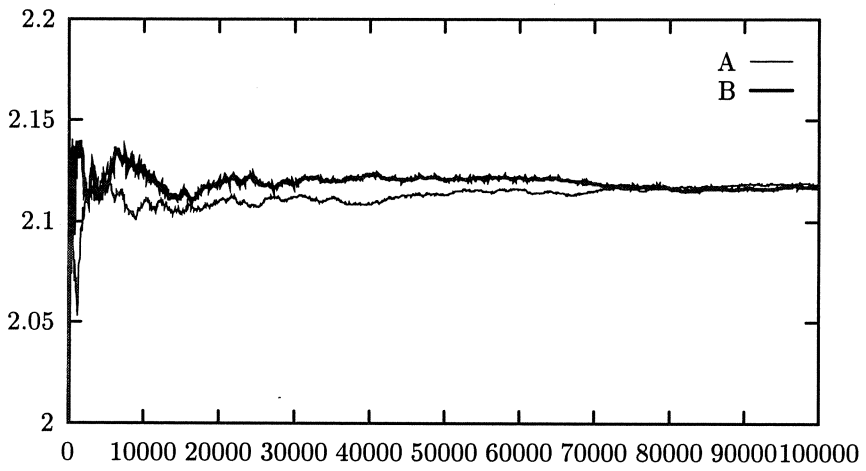


FIGURE 9. Evolution of the number of tokens in the place p_6 with two different initial markings. For A, we used the initial marking $(0,0,0,0,2,0,3,0)$. For B we used the initial marking $(1,1,1,1,1,1,1,0)$.

5.4 Open Systems With Several Initial Components

An input is slightly different from an initial component in the sense that it is not really part of the system but generated by an outside system which is considered unknown. No modelisation efforts have been made to describe more precisely the input. On the contrary an initial component is part of the system which is described and only represents the first part of the process. This is the reason why we distinguish these 2 cases in the following.

5.4.1 System With Several Initial SCMG

If our system has several initial components which are not all inputs, and if these components are really independent one with each other, the system is never stable [3], therefore an initial marking optimization may not be appropriate in this case since any initial modification may have an influence on the whole future of the system that does not couple with any stationary regime. If some dependency exists between the different initial components, then this could mean that there is a hidden dependency on an external phenomenon. In this case, we face a modelisation problem and the system was not correctly put under the form of a marked graph for our purpose.

5.4.2 System With Several Inputs

Several inputs are jointly ergodic and stationary (see the assumptions presented in the preliminaries). This can be interpreted as a common dependency on a preceding phenomenon. More precisely, we can see this system as a system with only one input as depicted in figure 10.

Suppose that a system has two input sequences $u(n)$ and $v(n)$. We construct a system with entry $w(n) = \min(u(n), v(n))$ and two places with temporizations $\max(u(n), v(n)) - v(n)$ to get $u(n)$ and $\max(u(n), v(n)) - u(n)$ to get $v(n)$.

The joint ergodicity of the sequences $u(n)$ and $v(n)$ implies that their ergodic shift θ is the same: $u(n) = u \circ \theta^n, v(n) = v \circ \theta^n$. Therefore, $\min(u(n), v(n)) = \min(u \circ \theta^n, v \circ \theta^n) = (\min(u, v)) \circ \theta^n$. Similarly, $\max(u(n), v(n)) - v(n) = (\max(u, v) - v) \circ \theta^n$ and $\max(u(n), v(n)) - u(n) = (\max(u, v) - u) \circ \theta^n$. The firing sequences of the transitions in the new system are stationary and ergodic, so the general theory of single input marked graphs applies to the new system. As for marking optimization, Theorem 5.6 applies when blocking the new input which is equivalent to blocking both inputs in the original system.

Indeed, in spite of the dependency between the firing sequences of the transitions that were added to the system, when we block the input, we also block these transitions in the system.

The coupling for these new transitions that is needed in the proof of theorem 5.6 becomes: $\phi_j^1(n) = \phi_j^2(n)$, for all the transitions j that have been introduced between the input and the system. This coupling is compatible with the dependencies of the firing sequences. Therefore, the proof of theorem 5.6 holds in this case.

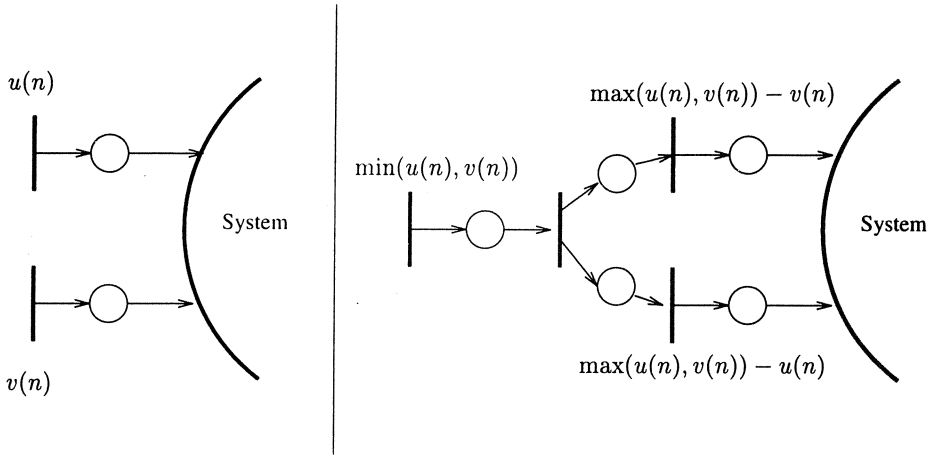


FIGURE 10. Transformation of a system with two inputs into a system with only one input. The firing times of the transitions are written on the figure.

6 Optimal Marking of Marked Graphs

In this section we discuss the practical interest of the markings M^* for parallel simulation of marked graphs.

6.1 Equational Simulations

Marked graphs can be very efficiently simulated using massive parallelism (see [4] and [13]). We describe briefly two kinds of equational simulation of a marked graph. In both cases we show that the complexity is linear in $L(M)$ and is close to the ideal case of a PRAM model.

The evolution of a Stochastic marked graph can be described by a linear system in the semi-field $\mathbb{R}(max, +)$:

$$X(n) = A(n)X(n-1).$$

The parallel algorithm developed in [4] uses these equations to compute the vector $X(n)$ of the firing times of the transitions. The computation of $A(n)$ involves $L(M) + 1$ operations of cost $log(|T|)$ each if made in parallel on a Connection Machine. Then, the matrix vector multiplication is done in parallel in $log(\overline{M}|T|)$ where $\overline{M} = \max_p M(p)$. The complexity depends heavily on the initial marking. It is of the form $O(n(L(M) + 1)l(|T|))$ with l being a logarithmic function. If the system fulfills the condition of theorems 5.6 or 5.3, then one can choose the initial marking which gives the best running time of the simulation, without altering the results of the simulation. This marking is a marking minimizing $L(M)$, i.e. a marking of the kind M^*.

This first algorithm only makes algebraic manipulations of the equations and ignores the underlying structure of the marked graph. A different approach uses

the topology and the marking of the marked graph to establish an order on the utilization of these equations. The transitions are distributed in the $L(M_0)$ classes , $C_k = \{t/C(t) = k\}$. All the equations associated with places in a same class are used in parallel. The simulation algorithm consists in:

```
for (n = 0 to N)
   for (i = 0 to L(M0))
      fire all the transitions in Ci.
```

In this approach, each transition is assigned to a different processor. "fire a transition t" means the application of the equation involving X_t which requires d operations, d being the entry degree of t (i.e. $\#^\bullet t$). The complexity is yet again linear in $L(M_0)$: $n(L(M_0)+1)\overline{d}$ where $\overline{d}$ is the max of all the degrees of the transitions. Once again, the marking M^* is the best initial marking of the system.

In both cases, the complexity of the algorithm is very close to the cost of any PRAM algorithm whose task graph is τ when started with a marking M^*. This seems to leave little hope for substantial improvement in this type of simulations (i.e. conservative) of marked graphs.

6.2 Applications

These results can be used in two different ways during a simulation of a given system. First, someone can *choose* an initial marking of type M^* that will satisfy the property $L(M^*) = L^*$, to start the simulation. Second, sometimes it is hard to find such a marking M^* by hand and furthermore, this marking may not correspond to the natural initial state of the system modeled by a marked graph. In these cases, the simulation may begin by a pre-computing step providing M^*. This initializing optimizer is available in the package MAGMAS© presented in [6] that provides simulation tools of marked graphs on a Connection Machine.

6.3 Experiments

We have run some experiments to show the improvement a marking optimization can provide on the speed of simulation. The jobshop model J depicted in figure 11 has an initial marking M_0 with $L(M_0) = 3$. A marking optimization gives the initial marking M^* depicted in figure 12 and $L(M^*) = 0$. The new jobshop model is called J^*.

The result is given in figure 13. The improvement in the running time with the optimized system is substantial (more than twice quicker). In any case, the initial step of a simulation that consists in optimizing the initial marking takes a negligible fraction of the total simulation time and provides a considerable speed-up.

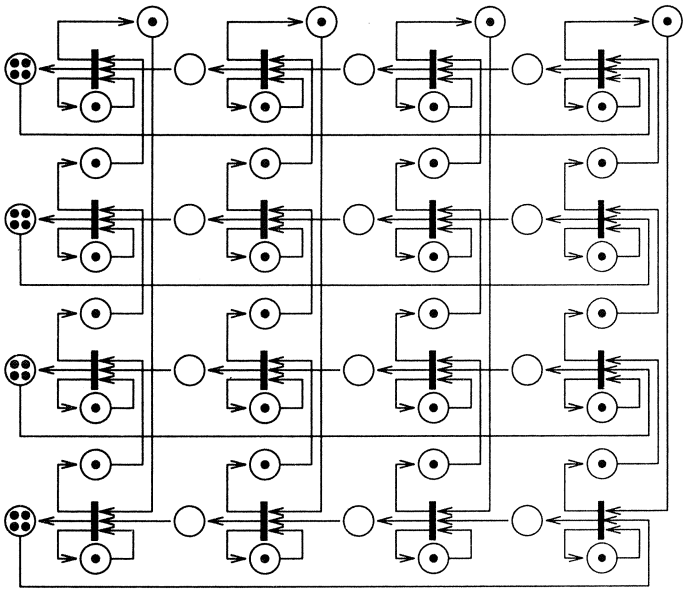


FIGURE 11. A jobshop model J with 4 types of machines and 4 types of products. There are 4 machines of each type with exponentially distributed firings of mean 1 and 4 pallets for carrying each type of product. The initial marking is not optimized. $L(M_0) = 3$.

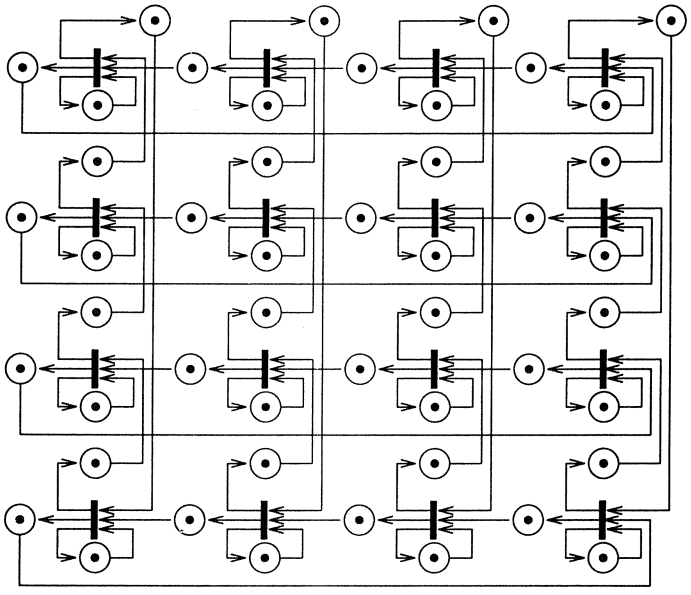


FIGURE 12. Jobshop J^* is the same as J but a different initial marking. We have computed the marking M^* for this system and $L(M^*) = 0$.

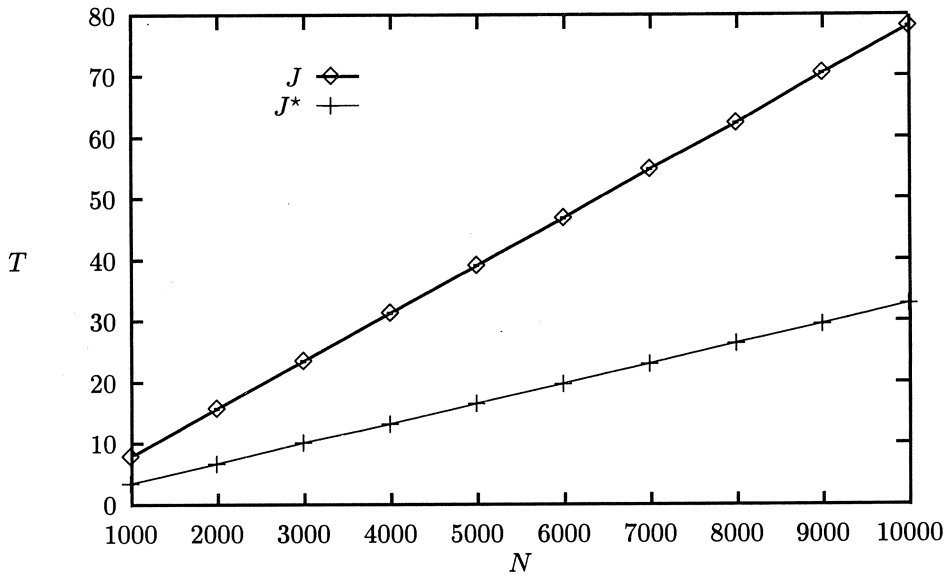


FIGURE 13. The performance of the optimized system is more than twice better than with the original one.

REFERENCES

1. E. Altman, M. Canales and A. Jean-Marie, *Stability and Limit Theorems in Discrete Event Systems*, In preparation.
2. F. Baccelli, *Ergodic Theory of Stochastic Petri Networks*, The Annals of Probability, Vol. 20, No. 1, 375-396, 1992.
3. F. Baccelli, G. Cohen, G.J. Olsder and J.P. Quadrat, *Synchronization and Linearity*. Wiley, 1992.
4. F. Baccelli and M. Canales, *Parallel Simulation of Stochastic Petri Nets using Recursive Equations*, ACM Transactions on Modeling and Computer Simulation, Vol. 3, No. 1, pp. 20-41, January 1993.
5. B. Berard and L. Thimonier, *On a Concurrency Measure*, 2nd I.S.C.I.S proceedings, Istanbul, 1987.
6. M. Canales, *Simulation parallèle de réseaux de Petri aléatoires utilisant des équations de récurrence*, Thèse de doctorat (in French), University of Nice, Sophia-Antipolis, to appear in Dec 1993.
7. J. Carlier and P. Chretienne, *Problèmes d'ordonnancement*, Masson (in French), 1988.
8. B. Charron-Bost, *Mesures de la concurrence et du parallélisme des calculs répartis*. Thèse de doctorat de l'université PARIS VII (in French), Sept 89.
9. G. Cohen, D. Dubois, J.P. Quadrat and M. Viot, *A Linear-System-Theoretic View of Discrete Event Processes and its Use for Performance Evaluation in Manufacturing*. IEEE Tr. Aut. Contr. ,Vol. 30, 210-220, 1985.

10. Y. Dallery, Z. Liu, D. Towsley, *Equivalence, Reversibility, Symmetry and Concavity Properties in Fork/Join Queuing Networks with Blocking.* INRIA Report 1267, 1990, to appear in J. ACM.
11. B. Dushnik and E.W. Miller, *Partially Ordered Sets.* Amer. Jour. of Math., 63:600-610, 1941.
12. S. Elmaghraby, *Activity Networks: Project Planning and Control by Networks Models.* Wiley, New York, 1977.
13. L. Finta, *Simulation parallèle de réseaux de Petri sur la Connection Machine*, Rapport de DEA I3S (in French), October 1992.
14. R. M. Fujimoto, *Parallel Discrete Event Simulation.* In E. A. MacNair, K.J. Musselman and P. Heidelberger, editors, Winter Simulation Conference, 1989.
15. D. Geniet, *AUTOMAF: un système de construction d'automates synchronisés et de calcul de mesure du parallélisme.* Thèse de doctorat de l'Université de Paris Sud, 1989.
16. M. Gondran and M. Minoux, *Graphes et Algorithmes.* Eyrolles editor (in French), 1979.
17. R. M. Karp, *A Characterization of the Minimum Cycle Mean in a Digraph.* Discrete Mathematics, vol 23, p. 309-311, 1978.
18. D.R. Jefferson and H.A. Sowizral, *Fast Concurrent Simulation using the Time Warp Mechanism.* Proc. of SCS Distributed Simulation Conference, p. 63-69, January 1993.
19. L. Lamport, *Time, Clocks and the Ordering of Events in a Distributed System.* Comm. of ACM, 21, 7:558-564, 1978.
20. S. Laftit, J.M. Proth and X.L. Xie, *Marking Optimization in Timed Event Graph.* Appl. and Th. of Petri Nets, 12th International Conference, p. 276-295, June 1991.
21. D.J. Leu and T.Murata, *Properties and Applications of the Token Distance Matrix of a Marked Graph.* Proc 1984 IEEE Int. Symp. Circuits Sys., Vol. 3, 1984.
22. J. Mairesse, *Products of Irreducible Random Matrices in the (Max,+) Algebra - Part I*, INRIA report No 1939, May 1993.
23. J. Mairesse, *Products of Irreducible Random Matrices in the (Max,+) Algebra - Part II*, INRIA report No 2072, Nov 1993.
24. T. Murata, *Petri Nets: Properties, Analysis and Applications.* Proc. of the IEEE, Vol 77, No 4, April 1989.
25. C. V. Ramamoorthy, G. S. Ho, *Performance Evaluation of Asynchronous Concurrent Systems Using Petri Nets.* IEEE Trans. on Software Engineering, Vol. SE-6, pp. 440-449, 1980.

Analytical Computation of Lyapunov Exponents in Stochastic Event Graphs*

Alain Jean-Marie
INRIA, Centre de Sophia Antipolis
2004, Route des Lucioles
06565 VALBONNE Cedex, France

We consider a simple stochastic model with synchronization, representing for instance the evolution of an elementary event graph. We develop a computational technique that leads in some cases to the complete, exact solution for both transient and stationary quantities of the model. The technique consists in translating the $(\max, +)$ linear system which describes the evolution of the event graph into a (standard) linear recurrence for the joint transforms of the state variables. In the continuous case, the technique is based on formulas of the calculus of the complex variable. In the discrete case, one may use similar formulas, as well as others involving linear algebra. These exact results can be used as a basis for evaluating the relative merits of several bounding schemes.

1 Introduction

The study reported here was initially motivated by the paper of Baccelli and Konstantopoulos [3]. In that paper, the authors give a methodology to derive bounds for the *cycle time* in a class of stochastic discrete event systems described by evolution equations involving addition and maximization. Our aim is to obtain the *exact* solution for models in this class, and to use the solution as a testbed for evaluating the accuracy of these bounds. We therefore study the simplest model in this class, represented in figure 1. It may be interpreted as the infinite and periodic task graph describing the execution of an elementary cyclic parallel program on two processors. It may also represent the evolution of a stochastic Petri Net (with a particular convention concerning the handling of tokens).

Let $\{\sigma_{ij}(n)\}_1^\infty$, $i,j \in \{1,2\}$ be four independent sequences of i.i.d random variables (RVs), hereafter called "input sequences". We are interested in solving the stochastic recurrence:

$$\begin{aligned} X_1(n) &= \max\{X_1(n-1)+\sigma_{11}(n), X_2(n-1)+\sigma_{21}(n)\} \\ X_2(n) &= \max\{X_1(n-1)+\sigma_{12}(n), X_2(n-1)+\sigma_{22}(n)\} \end{aligned} \tag{1}$$

*Supported by the European Grant BRA-QMIPS of CEC DG XIII.

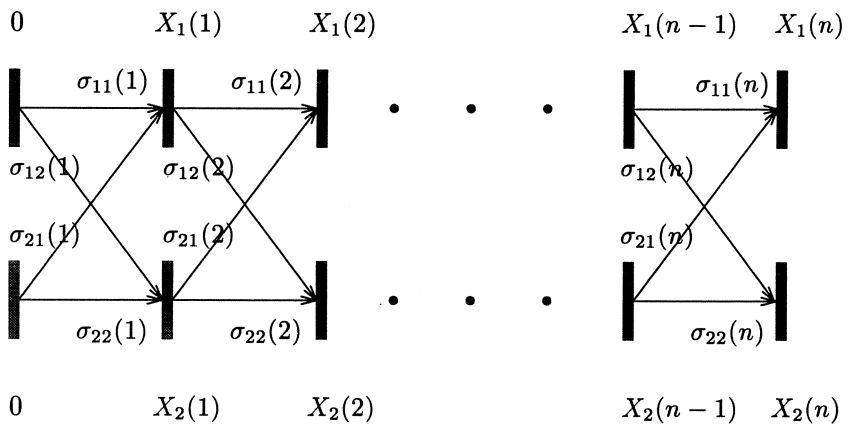


FIGURE 1. Representation of the stochastic recurrence

with initial conditions $X_1(0) = X_2(0) = 0$.

This system is the simplest linear recurrence in the semi-field $(\max, +)$ which has the "strong connectivity" (or irreducibility) property. This property essentially means that the sequence $\{X_2(n)\}$ depends on the sequence $\{X_1(n)\}$ and conversely. On the other hand, if in (1) we had $\sigma_{21}(n) = -\infty$ for all n, then the system is known to model the evolution of a $GI/GI/1$ queue. In that case, the sequence $\{X_1(n)\}$ does not depend on the sequence $\{X_2(n)\}$.

The general theory of such recurrences is developed in [1], and it is known that under natural conditions on the input sequences, the RVs $X_1(n)/n$ and $X_2(n)/n$ both converge almost surely to a constant, γ, often referred to as the "Lyapunov Exponent" (anciently, "Kingman constant") of the system. This constant may be interpreted in terms of *cycle times* and *throughput*. It is an essential performance measure of the discrete event system modelled by (1), and as such, its computation or estimation has recently received much attention in the literature.

In [10], a method is proposed to obtain the cycle time of the system. It is based on the fact that the variables $\{X_1(n) - X_2(n)\}_n$ form a Markov Chain. The cycle time γ can be expressed in terms of its stationary distribution. In the case where the input sequences are not discrete, this involves the solution of integral equations. In [10], computations are carried out when all four sequences have the same distribution (the "totally symmetric case"), and when this distribution is exponential, Bernoulli or uniform.

The investigation of the "transients" of the system is interesting from many points of view. First, it allows to appreciate the speed of the convergence of the distributions of $X_i(n)/n$ $(i = 1, 2)$ and $X_1(n) - X_2(n)$ to their respective limits. Secondly, one may use the successive distributions of $X_1(n)$ to compute bounds, either by using known "oversynchronization" and "undersynchroniza-

tion" techniques for task graphs [2] with the representation of figure 1, or following the method exposed in [3]. The comparison of these different approximation schemes, based on exact results obtained in this paper, is currently under development.

In a first part of this paper, we revisit the method of Resing *et al.* for discrete distributions, and show how the joint generating functions of the state vector can be computed recursively by a linear recurrence (theorem 2.1). As a corollary, we obtain two distinct ways of computing the Lyapunov exponent of the system.

In a second part, we propose a method to compute the successive joint distributions of the couple $(X_1(n), X_2(n))$ in the case where the distributions are continuous. We prove that the joint Laplace transforms are given by a linear recurrence (theorem 3.1).

The rest of the paper gives examples of practical computations in simple cases. We study the case of exponential random variables in the "symmetric" (section 4.1) and "semi-symmetric" (section 4.2) cases. In these cases, the linear recurrence can be expressed as a simpler recurrence on a small number of polynomials (lemma 4.1 and (28)). This method is then developed for discrete distributions with infinite support, for which the case of Bernoulli RV's with arbitrary parameters is solved (section 5).

2 The Linear Recurrence in the Discrete Case

Assume in this section that $\sigma_{1,1}, \sigma_{1,2}, \sigma_{2,1}$ and $\sigma_{2,2}$ all have a discrete distribution. In this case, we shall refine the technique proposed by Resing *et al.* in [10] and obtain two ways to compute the cycle time γ and the asymptotic variance σ of the system.

Let, for all $n \geq 0$,

$$Z(n) = X_1(n) - X_2(n), \qquad \eta(n) = \min\{X_1(n), X_2(n)\} .$$

It is known [10] that under the independence assumption of the input sequences, $\mathbf{Z} = \{Z(n)\}_n$ is a homogeneous Markov chain with state space in $\mathbb{Z}$. If the distributions of the $\sigma_{i,j}$ are all bounded, the state space of this Markov chain is bounded as well.

It is easy to show that:

$$\begin{cases} X_1(n+1) &= \eta(n) + \max\{Z(n)^+ + \sigma_{11}(n+1), Z(n)^- + \sigma_{21}(n+1)\} \\ X_2(n+1) &= \eta(n) + \max\{Z(n)^+ + \sigma_{12}(n+1), Z(n)^- + \sigma_{22}(n+1)\} . \end{cases} \quad (2)$$

and

$$\forall n \geq 0,\ \eta(n+1) = \eta(n) + \min \begin{array}{l} \{Z(n)^- + \max(\sigma_{11}(n+1), \sigma_{12}(n+1)), \\ Z(n)^+ + \max(\sigma_{21}(n+1), \sigma_{22}(n+1))\} , \end{array} \quad (3)$$

where, by notation: $x^+ = max(x, 0)$ and $x^- = -min(x, 0)$.

Finally, let $\Phi_n(s,t) = \mathbb{E}(s^{X_1(n)} t^{X_2(n)})$.

THEOREM 2.1 *The sequence Φ_n is given by:*

$$\Phi_n(s,t) \;=\; \sum_{k\in\mathbb{Z}} P_n^{(k)}(st)\; s^{k^+}t^{k^-}\;, \tag{4}$$

where $P_n^{(k)}(x) = \mathbb{E}(x^{\eta(n)}\mathbf{1}_{\{Z(n)=k\}})$. *Moreover, the sequence of vectors* $\mathbf{p}_n(x) = (P_n^{(k)}(x))_{k\in\mathbb{Z}}$ *satisfies the linear recurrence:*

$$\mathbf{p}_{n+1}(x) \;=\; \mathbf{M}(x)\;\mathbf{p}_n(x)\;, \tag{5}$$

with

$$M_{jk}(x) \;= \tag{6}$$

$$\mathbb{E}(x^{\min\{k^-+\max(\sigma_{11}(n),\sigma_{12}(n)),k^++\max(\sigma_{21}(n),\sigma_{22}(n))\}}\mathbf{1}_{\{Z(n)=j\}}|Z(n-1)=k)\;.$$

Note that the definition of the matrix $\mathbf{M}(x)$ in (6) does not depend on $n \geq 1$ because the chain $\mathbf{Z}$ is homogeneous.

PROOF Decomposing Φ_n according to the value of $Z(n)$ and using the variable $\eta(n)$, one obtains:

$$\begin{aligned}\Phi_n(s,t) \;&=\; \sum_{k\in\mathbb{Z}} \mathbb{E}(s^{\eta(n)+k^+}t^{\eta(n)+k^-}\mathbf{1}_{\{Z(n)=k\}})\\ &=\; \sum_{k\in\mathbb{Z}} s^{k^+}t^{k^-}\,\mathbb{E}((st)^{\eta(n)}\mathbf{1}_{\{Z(n)=k\}})\;.\end{aligned}$$

This proves (4). Assume now that $n \geq 1$. Conditioning on $Z(n-1)$, and making use of (3) and the Markov property, one has:

$$P_n^{(j)}(x) =$$

$$\sum_{i\in\mathbb{N}}\sum_{k\in\mathbb{Z}} \mathbb{E}(x^{\eta(n-1)+\min\{k^-+\max(\sigma_{11}(n),\sigma_{12}(n)),k^++\max(\sigma_{21}(n),\sigma_{22}(n))\}}\mathbf{1}_{\{Z(n+1)=j\}}$$

$$\begin{aligned}&|Z(n-1)=k,\eta(n-1)=i)\;\mathbb{P}(Z(n-1)=k,\eta(n-1)=i)\\ &=\sum_{i\in\mathbb{N}}\sum_{k\in\mathbb{Z}} x^i m_{jk}\mathbb{P}(Z(n-1)=k,\eta(n-1)=i)\\ &=\sum_{k\in\mathbb{Z}} m_{jk}\mathbb{E}(x^{\eta(n-1)}|Z(n-1)=k)\mathbb{P}(Z(n-1)=k)\;,\end{aligned}$$

which proves (5).

One immediately notices that the matrix $\mathbf{M}(1) = \boldsymbol{\Pi}$ is the transition matrix of the Markov chain $\mathbf{Z}$ and that $\pi_n = \mathbf{p}(1)$ is the probability distribution vector of $Z(n)$.

The representation of theorem 2.1 allows to compute transient and stationary quantities of the system in two (seemingly) distinct ways: one using the stationary distribution of $\mathbf{Z}$ (theorem 2.2), and one using uniquely the characteristic polynomial of $\mathbf{M}(x)$ (theorem 2.3).

Let $d(n) = X_1(n) - X_1(n-1)$. the quantity $d(n)$ is therefore the increase of X_1 at the n–th step, and we expect naturally that when the system is stationary, its mean value will be the growth rate of the system (*i.e.* its Lyapunov exponent), that is:

$$\gamma \;=\; \mathbb{E}_\pi(d(n)) \;.$$

We are going to analyze some properties of the sequence $\{d(n)\}_n$. The properties of the variables $X_2(n) - X_2(n-1)$ will follow easily.

THEOREM 2.2

i/ We have:

$$\mathbb{E}d(n) \;=\; \mathbf{h}\,\pi_{n-1} \;=\; \mathbf{h}\,\boldsymbol{\Pi}^{n-1}\pi_0 \;. \tag{7}$$

where $\mathbf{h}$ *is the vector:*

$$\mathbf{h} \;=\; \mathbf{e}\mathbf{M}'(1) \;+\; \mathbf{f}(\boldsymbol{\Pi} - \mathbf{I}) \;,$$

with $\mathbf{e} = (1)_{k\in\mathbb{Z}}$ *and* $\mathbf{f} = (k^+)_{k\in\mathbb{Z}}$.

ii/ The covariances of the sequence $\{d(n)\}$ *are given by:*

$$n \geq 2 : \; \mathbb{E}(d(n)d(1)) = \mathbf{h}\boldsymbol{\Pi}^{n-2}(\mathbf{M}'(1) + \boldsymbol{\Pi}\mathbf{F}\boldsymbol{\Pi} - \mathbf{F})\pi_0 \tag{8}$$

$$\mathbb{E}(d(1)^2) = E(d(1)) \;+\; (\mathbf{e}\mathbf{M}''(1) + 2\mathbf{f}\mathbf{M}'(1) - 2\mathbf{h}\mathbf{F})\pi_0 \;, \tag{9}$$

where $\mathbf{F}$ *is the diagonal matrix with* $\mathbf{F}_{ii} = i^+$.

iii/ If $\mathbf{Z}$ *is stationary and ergodic with stationary distribution* π, *one has:*

$$\lim_{n\to\infty} \mathbb{E}d(n) \;=\; \gamma \;=\; \mathbf{e}\mathbf{M}'(1)\pi \;,$$

and, in distribution:

$$\frac{X_1(n) - n\gamma}{\sqrt{n}} \;\xrightarrow[n\to\infty]{w}\; Y,$$

with Y *normally distributed with mean 0 and variance* σ *given by:*

$$\sigma^2 \;=\; \gamma - 3\gamma^2 \;+\; \mathbf{e}\big[\mathbf{M}''(1) + 2\mathbf{M}'(1)(\mathbf{I} - \boldsymbol{\Pi} - \pi\mathbf{e})^{-1}\mathbf{M}'(1)\big]\;\pi \;. \tag{10}$$

PROOF From Theorem 2.1, we have:

$$\mathbb{E}(s^{X_1(n)}) \;=\; \mathbf{f}(s)\mathbf{p}_n(s) \;, \tag{11}$$

where $\mathbf{f}(s) = (s^{k^+})_{k\in\mathbb{Z}}$. The vectors $\mathbf{f}(s)$ and $\mathbf{p}_n(s)$ are obviously differentiable with respect to s. Moreover, $\mathbf{f}(1) = \mathbf{e}$ and $\mathbf{f}'(1) = \mathbf{f}$. Therefore:

$$EX_1(n) \;=\; \mathbf{f}\pi_n \;+\; \mathbf{e}\mathbf{p}_n'(1) \;.$$

Using the relation: $\mathbf{p}_n(s) = \mathbf{M}(s)\mathbf{p}_{n-1}(s)$, we have:

$$\mathbf{p}'_n(1) \;=\; \mathbf{M}'(1)\pi_{n-1} \;+\; \boldsymbol{\Pi}\mathbf{p}'_{n-1}(1)\;. \tag{12}$$

It follows that for $n \geq 1$:

$$EX_1(n) - EX_1(n-1) \;=$$

$$\mathbf{f}(\boldsymbol{\Pi}\pi_{n-1} - \pi_{n-1}) \;+\; \mathbf{e}\mathbf{M}'(1)\pi_{n-1} + \mathbf{e}(\boldsymbol{\Pi} - I)\mathbf{p}'_{n-1}(1)\;.$$

But $\mathbf{e}\boldsymbol{\Pi} = \mathbf{e}$, and this relation rewrites as (7). This proves i/.

The covariance of $d(1)$ and $d(n)$ is:

$$\mathbb{E}(d(1)d(n)) \;=\; \mathbb{E}(d(n)X_1(1)) \;-\; \mathbb{E}(d(n)X_1(0))\;.$$

For all $n \geq 2$, one has:

$$\mathbb{E}(d(n)X_1(1)) \;=\; \mathbf{h}\boldsymbol{\Pi}^{n-2}\mathbb{E}(\pi_1 X_1(1))\;.$$

But the k-th component of $\mathbb{E}(\pi_1 X_1(1))$ is $\mathbb{E}(\mathbf{1}_{\{Z(1)=k\}}X_1(1)) = P^{(k)\prime}_1(1) + k^+P^{(k)}_1(1)$, from the definition of $P^{(k)}_n(x)$. Therefore,

$$\mathbb{E}(\pi_1 X_1(1)) \;=\; \mathbf{p}'_n(1) \;+\; \mathbf{F}\pi_1,$$

and likewise,

$$\mathbb{E}(\pi_1 X_1(0)) \;=\; \mathbf{p}'_n(0) \;+\; \mathbf{F}\pi_0.$$

Computing the difference and using $\mathbf{p}'_1(1) = \mathbf{M}'(1)\pi_0 + \boldsymbol{\Pi}\mathbf{p}'_0(1)$, one obtains (8).

Let us compute the variance of $d(1)$. Differentiating (11) twice, we have:

$$E[X_1(n)(X_1(n)-1)] \;=\; \mathbf{g}\pi_n \;+\; 2\mathbf{f}\mathbf{p}'_n(1) \;+\; \mathbf{e}\mathbf{p}''_n(1)\;,$$

with $\mathbf{g} = \mathbf{f}''(1) = \mathbf{0}$. Moreover, $\mathbf{p}''_n(1) = \mathbf{M}'(1)\pi_n + 2\mathbf{M}'(1)\mathbf{p}'_{n-1}(1) + \boldsymbol{\Pi}\mathbf{p}''_{n-1}(1)$. Then, writing:

$$E[d(1)^2] \;=\; E[X_1(1)^2] - E[X_1(0)^2] - 2E[d(1)X_1(0)]\;,$$

and using previous calculations, one obtains:

$$\begin{aligned} E[d(1)^2] \;&=\; Ed(1) \;+\; 2\mathbf{f}(\mathbf{M}'(1)\pi_0 + (\boldsymbol{\Pi} - \mathbf{I})\mathbf{p}'_0(1)) \\ &\quad + \mathbf{e}(\boldsymbol{\Pi} - \mathbf{I})\mathbf{p}''_0(1) \;-\; 2\mathbf{h}(\mathbf{F}\pi_0 + \mathbf{p}'_0(1)) \\ &=\; Ed(1) \;+\; 2(\mathbf{f}\mathbf{M}'(1) - \mathbf{h}\mathbf{F})\pi_0 \;+\; 2(\mathbf{f}(\boldsymbol{\Pi} - \mathbf{I}) - \mathbf{h})\mathbf{p}'_0(1)\;, \end{aligned}$$

because $\mathbf{e}(\boldsymbol{\Pi} - \mathbf{I}) = 0$. This finally rewrites as (9).

If $\mathbf{Z}$ has a stationary distribution π, then for all non-zero vectors π_0, we have: $\lim_{n\to\infty} \boldsymbol{\Pi}^n\pi_0 = \pi$ (see *e.g.* [11]). Therefore, $\lim_{n\to\infty} \mathbb{E}(d(n)) = \mathbf{h}\pi = \mathbf{e}\mathbf{M}'(1)\pi$ (because $\boldsymbol{\Pi}\pi = \pi$). The validity of the central limit theorem for $X_1(n)$ has been proved in [10]. In this paper, the asymptotic variance is given by the formula:

$$\sigma^2 \;=\; \mathbb{E}_\pi(d(1)^2) + 2\sum_{l=2}^{\infty} \mathbb{E}_\pi[(d(1)-\gamma)(d(l)-\gamma)]\;,$$

where $\mathbb{E}_\pi$ denotes the expectation given that the distribution of $Z(0)$ is given by π (that is, $\mathbf{Z}$ is stationary). Assume first that $n \geq 2$. From (8), and remembering that $\lim_{n\to\infty} \boldsymbol{\Pi}^n = \pi\mathbf{e}$, one writes:

$$\begin{aligned}
&\mathbb{E}_\pi[(d(1)-\gamma)(d(n)-\gamma)] = \mathbb{E}_\pi(d(1)d(n)) - \gamma^2 \\
&= \mathbf{h}[\boldsymbol{\Pi}^{n-2} - \pi\mathbf{e} + \pi\mathbf{e}][\mathbf{M}'(1) + (\mathbf{I}-\boldsymbol{\Pi})\mathbf{F}]\pi - \gamma^2 \\
&= \mathbf{h}[\boldsymbol{\Pi}^{n-2} - \pi\mathbf{e}]\mathbf{M}'(1)\pi + \mathbf{h}[\boldsymbol{\Pi}^{n-2} - \pi\mathbf{e}](\mathbf{I}-\boldsymbol{\Pi})\mathbf{F}\pi + \gamma^2 - \gamma^2 \\
&= \mathbf{h}[\boldsymbol{\Pi}^{n-2} - \pi\mathbf{e}]\mathbf{M}'(1)\pi + \mathbf{h}[\boldsymbol{\Pi}^{n-2} - \boldsymbol{\Pi}^{n-1}]\mathbf{F}\pi \\
&= \mathbf{e}\mathbf{M}'(1)[\boldsymbol{\Pi}^{n-2} - \pi\mathbf{e}]\mathbf{M}'(1)\pi + \mathbf{f}[\boldsymbol{\Pi}^{n-1} - \boldsymbol{\Pi}^{n-2}]\mathbf{F} \\
&+\mathbf{h}[\boldsymbol{\Pi}^{n-2} - \boldsymbol{\Pi}^{n-1}]\mathbf{F}\pi.
\end{aligned}$$

It follows that:

$$\sum_{l=2}^{\infty} \mathbb{E}_\pi[(d(1)-\gamma)(d(l)-\gamma)] \;=$$

$$\left(\mathbf{e}\mathbf{M}'(1)(\sum_{n=0}^{\infty}[\boldsymbol{\Pi}^n - \pi\mathbf{e}])\mathbf{M}'(1) + \mathbf{f}(\pi\mathbf{e}-\mathbf{I})\mathbf{F} + \mathbf{h}(\mathbf{I}-\pi\mathbf{e})\mathbf{F}\right)\pi\;.$$

To compute the series, notice that if $n > 0$, $\boldsymbol{\Pi}^n - \pi\mathbf{e} = (\boldsymbol{\Pi} - \pi\mathbf{e})^n$. Therefore, $\sum_{n=0}^{\infty}[\boldsymbol{\Pi}^n - \pi\mathbf{e}] = (\mathbf{I} - \boldsymbol{\Pi} - \pi\mathbf{e})^{-1} - \pi\mathbf{e}$. The above expression then reduces to:

$$\sum_{l=2}^{\infty} \mathbb{E}_\pi[(d(1)-\gamma)(d(l)-\gamma)] = \mathbf{e}\mathbf{M}'(1)(\mathbf{I} - \boldsymbol{\Pi} - \pi\mathbf{e})^{-1}\mathbf{M}'(1)\pi$$

$$+\mathbf{h}(\mathbf{I}-\pi\mathbf{e})\mathbf{F}\pi - \gamma^2.$$

Together with (9), one obtains (10).

THEOREM 2.3 *Let* $P(z,x) = Det(\mathbf{M}(x) - z\mathbf{I})$ *be the characteristic polynomial of* $\mathbf{M}(x)$. *Let:*

$$P_x = \frac{\partial P}{\partial x}(1,1),\quad P_z = \frac{\partial P}{\partial z}(1,1),\quad P_{xx} = \frac{\partial^2 P}{\partial x^2}(1,1),$$

$$P_{xz} = \frac{\partial^2 P}{\partial x \partial z}(1,1),\ \textit{and}\ P_{zz} = \frac{\partial^2 P}{\partial z^2}(1,1).$$

Then:

$$\gamma \;=\; -\frac{P_x}{P_z}\;, \tag{13}$$

and

$$\sigma \;=\; \gamma \;-\; \gamma^2 \;-\; \frac{1}{P_z}\left(P_{xx} + 2\gamma P_{xz} + \gamma^2 P_{zz}\right)\;. \tag{14}$$

PROOF Let $X_1^*(n)(s) = \mathbb{E}(s^{X_1(n)})$. Using, for instance, the generating matrix $\sum_{n=0}^{\infty} t^n \mathbf{M}(x)^n$, one can show that for all $n \geq 0$ and in a neighborhood V_1 of $s = 1$, one has (see [11, p. 9]):

$$X_1^*(n)(s) \;=\; \Phi_n(s,1) \;=\; \sum_{l=1}^{L} \sum_{j=1}^{m_l - 1} n^j \alpha_{lj}(s) \lambda_l^n(s) \;,$$

where $\lambda_1, \ldots, \lambda_L$ are the L distinct eigenvalues of $\mathbf{M}(s)$, m_l the multiplicity of λ_l and the α_{lj} are functions independent of n, analytic in V_1.

The Perron-Frobenius eigenvalue of $\mathbf{M}(s)$, say $\lambda_1(s)$, is real, isolated ($m_1 = 1$) and less than 1, because $\mathbf{M}(s)$ is substochastic for $s \in [0,1]$. Moreover, $\lambda_1(1) = 1$ and there exists $\rho < 1$ such that

$$\forall s \in V_1,\ \forall l, 2 \leq l \leq L, \qquad |\lambda_l(s)| < \rho \;.$$

Consequently,

$$X_1^*(n)(s) \;=\; \lambda_1^n(s)\ (1 + o(\rho^n)) \;.$$

A straightforward asymptotic expansion in $X_1^*(n)(e^{-s/\sqrt{n}})$ allows to conclude that as $n \to \infty$,

$$\frac{X_1(n) - n\gamma}{\sigma\sqrt{n}} \xrightarrow{w} Y,$$

with $Y \sim \mathcal{N}(0,1)$ and with

$$\gamma = \lambda_1'(1) \quad \text{and} \quad \sigma^2 = \lambda_1''(1) + \gamma - \gamma^2 \;.$$

The derivatives of $\lambda_1(s)$ at $s = 1$ can hopefully be computed without an explicit expression for this function. It suffices to note that $P(z,s) = \prod_{l=1}^{L} (z - \lambda_l(s))^{m_l}$, so that:

$$P_z \;=\; \sum_{l=1}^{L} m_l (z - \lambda_l(s))^{m_l - 1} \prod_{j \neq l} (z - \lambda_j(s))^{m_j} \;|_{s=1, z=1} \;=\; \prod_{j \neq 1} (z - \lambda_j(1))$$

and

$$P_x \;=\; -\sum_{l=1}^{L} \lambda_l'(s) m_l (z - \lambda_l(s))^{m_l - 1} \prod_{j \neq l} (z - \lambda_j(s))^{m_j} \;|_{s=1, z=1} \;=\; -\lambda_1'(1) P_z \;.$$

The expression for $\lambda_1'(1)$ follows. A similar calculation yields $\lambda_1''(1)$. □

3 THE CONTINUOUS CASE

For the moment, we do not assume anything particular on the distribution of the input sequences. For $i, j \in 1, 2$, let us denote $S_{ij}(x) = \mathbb{P}(\sigma_{ij}(n) \leq x)$. Let, for all $n \geq 0$:

$$F_n(x,y) = P(X_1(n) \leq x,\ X_2(n) \leq y),$$

and

$$\Psi_n(s,t) = E(e^{-sX_1(n)-tX_2(n)}) = \int_{\mathbb{R}}\int_{\mathbb{R}} e^{-sx-ty} dF_n(x,y) \ .$$

This Laplace Transform is defined *a priori* on the domain $i\mathbb{R}\times i\mathbb{R}$. If the random variables $X_1(n)$ and $X_2(n)$ are known to be positive (the most common case), the domain can be extended to $\{\Re(s) \geq 0\} \times \{\Re(t) \geq 0\}$. It is well known that

$$\int_{\mathbb{R}}\int_{\mathbb{R}} e^{-sx-ty} F_n(x,y)dxdy = \frac{1}{st}\Psi_n(s,t) \ . \tag{15}$$

The initial values of these functions are naturally $F_0(x,y) = \mathbf{1}_{\{x,y\geq 0\}}$ and $\Psi_0(s,t) = 1$.

From (1), one has:

$$\begin{aligned} F_{n+1}(x,y) &= \mathbb{P}(X_1(n) + \sigma_{11}(n+1) \leq x, X_2(n) + \sigma_{21}(n+1) \leq x, \\ &\qquad X_1(n) + \sigma_{12}(n+1) \leq y, X_2(n) + \sigma_{22}(n+1) \leq y) \\ &= \int_{\mathbb{R}}\int_{\mathbb{R}} \mathbb{P}(\sigma_{11}(n+1) \leq x-u, \sigma_{21}(n+1) \leq x-v, \\ &\qquad \sigma_{12}(n+1) \leq y-u, \sigma_{22}(n+1) \leq y-v) dF_n(u,v) \\ &= \int_{-\infty}^{\inf(x,y)}\int_{-\infty}^{\inf(x,y)} S(x,y,u,v)dF_n(u,v) \ , \end{aligned} \tag{16}$$

where $S(x,y,u,v) = S_{11}(x-u)S_{21}(x-v)S_{12}(y-u)S_{22}(y-v)$.

Obeying to a standard reflex in such a context, we turn to Laplace transforms, hoping to obtain a more appealing recurrence relation. From (15) and (16), we have:

$$\begin{aligned} &\frac{1}{st}\Psi_{n+1}(s,t) \\ &= \int_{\mathbb{R}}\int_{\mathbb{R}} e^{-sx-ty} \int_{\mathbb{R}}\int_{\mathbb{R}} \mathbf{1}_{\{u\leq \inf(x,y)\}}\mathbf{1}_{\{v\leq \inf(x,y)\}} S(x,y,u,v)dF_n(u,v)dxdy \\ &= \int_{\mathbb{R}}\int_{\mathbb{R}} e^{-sx-ty} \int_{\mathbb{R}}\int_{\mathbb{R}} \left(\mathbf{1}_{\{u\leq v\leq \inf(x,y)\}} + \mathbf{1}_{\{v<u\leq \inf(x,y)\}}\right) S(x,y,u,v)dF_n(u,v)dxdy \\ &= \int_{\mathbb{R}}\int_{\mathbb{R}} \mathbf{1}_{\{u>v\}} \int_u^\infty \int_u^\infty e^{-sx-ty} S(x,y,u,v)dxdydF_n(u,v) \\ &\quad + \int_{\mathbb{R}}\int_{\mathbb{R}} \mathbf{1}_{\{u\leq v\}} \int_v^\infty \int_v^\infty e^{-sx-ty} S(x,y,u,v)dxdydF_n(u,v) \\ &= \int_{\mathbb{R}}\int_{\mathbb{R}} \mathbf{1}_{\{u>v\}} e^{-u(s+t)} \int_{\mathbb{R}^+}\int_{\mathbb{R}^+} e^{-sx-ty} S(x+u,y+u,u,v)dxdydF_n(u,v) \\ &\quad + \int_{\mathbb{R}}\int_{\mathbb{R}} \mathbf{1}_{\{u\leq v\}} e^{-v(s+t)} \int_{\mathbb{R}^+}\int_{\mathbb{R}^+} e^{-sx-ty} S(x+v,y+v,u,v)dxdydF_n(u,v) \end{aligned} \tag{17}$$

Defining the functions:

$$
\begin{aligned}
&A_n(s,t) \\
&= \int_{\mathbb{R}}\int_{\mathbb{R}} \mathbf{1}_{\{u>v\}} e^{-u(s+t)} \int_{\mathbb{R}}\int_{\mathbb{R}} S(x+u,y+u,u,v) e^{-sx-ty} dx dy dF_n(u,v) \\
&B_n(s,t) \\
&= \int_{\mathbb{R}}\int_{\mathbb{R}} \mathbf{1}_{\{v\geq u\}} e^{-v(s+t)} \int_{\mathbb{R}}\int_{\mathbb{R}} S(x+v,y+v,u,v) e^{-sx-ty} dx dy dF_n(u,v) \\
&= \int_{\mathbb{R}}\int_{\mathbb{R}} \mathbf{1}_{\{u\geq v\}} e^{-u(s+t)} \int_{\mathbb{R}}\int_{\mathbb{R}} S(x+u,y+u,v,u) e^{-sx-ty} dx dy dF_n(v,u) \ ,
\end{aligned}
$$

equation (17) reduces to:

$$
\frac{1}{st}\Psi_{n+1}(s,t) \ = \ A_n(s,t) + B_n(s,t) \ . \tag{18}
$$

Note that the definitions of A_n and B_n are slightly asymmetric. This is due to the fact that F_n may in all generality have a mass on the diagonal $u = v$. A more symmetric approach would be to isolate this diagonal and introduce three functions, as will be done in the discrete case (see (38) in section 5 and appendix A). In the cases we shall analyze in the following, this problem does not arise, and the decomposition (18) will be sufficient.

The rest of this section is devoted to the proof of the following theorem:

THEOREM 3.1 *There exists a function $K(z,s,t)$ such that, for all $n \geq 1$:*

$$
\Psi_{n+1}(s,t) \ = \ \frac{1}{2i\pi}\int_{i\mathbb{R}} K(z,s,t)\ \Psi_n(s+t+z,-z)\ dz \ .
$$

The function K is a 4-linear functional of the Laplace transforms of the distributions of the RV's σ_{ij}, $1 \leq i,j \leq 2$.

PROOF The inner integral of A and B equals:

$$
\int_{\mathbb{R}} S_{11}(x) S_{21}(x+u-v) e^{-sx} dx \int_{\mathbb{R}} S_{12}(y) S_{22}(y+u-v) e^{-ty} dy \ . \tag{19}
$$

Each of these terms is a Laplace transform. The first one is the Laplace transform of the r.v. $\max\{\sigma_{11}, \sigma_{21}+u-v\}$. Denote this r.v. by S, and let S^* be its Laplace transform. According to lemma A.3 (see also [8, p. 128]), we have:

$$
\begin{aligned}
S^*(s) \ = \ &- \lim_{z\to 0, \Re(z)>0} \Big[\frac{1}{2i\pi}\int_{i\mathbb{R}} \sigma_{11}^*(s+y)\sigma_{21}^*(-y) e^{y(u-v)} \frac{dy}{y-z} \\
&+ \frac{1}{2i\pi}\int_{i\mathbb{R}} \sigma_{11}^*(-y)\sigma_{21}^*(s+y) e^{-(s+y)(u-v)} \frac{dy}{y-z}\Big] \ .
\end{aligned}
$$

If follows that equation (19) can be rewritten as

$$
\lim_{z_1,z_2\to 0, \Re(z_1),\Re(z_2)>0} \Big(\frac{1}{2i\pi}\Big)^2 \int_{i\mathbb{R}}\int_{i\mathbb{R}} H(s,t,y_1,y_2) e^{y_1(u-v)} e^{y_2(u-v)}
$$

$$\frac{dy_1}{y_1 - z_1}\frac{dy_2}{y_2 - z_2} ,$$

with

$$\begin{aligned} H(s,t,y_1,y_2) &= \sigma^*_{11}(s+y_1)\sigma^*_{21}(-y_1)\sigma^*_{12}(s+y_2)\sigma^*_{22}(-y_2)e^{(y_1+y_2)(u-v)} \\ &+\sigma^*_{11}(s+y_1)\sigma^*_{21}(-y_1)\sigma^*_{12}(-y_2)\sigma^*_{22}(s+y_2)e^{(y_1-y_2-t)(u-v)} \\ &+\sigma^*_{11}(-y_1)\sigma^*_{21}(s+y_1)\sigma^*_{12}(s+y_2)\sigma^*_{22}(-y_2)e^{(y_2-y_1-s)(u-v)} \\ &+\sigma^*_{11}(-y_1)\sigma^*_{21}(s+y_1)\sigma^*_{12}(-y_2)\sigma^*_{22}(s+y_2)e^{-(y_1+y_2+s+t)(u-v)} . \end{aligned}$$

Integrating the exponential term of the first term on the domain $\{u > v\}$ against $e^{-(s+t)u}dF_n(u,v)$, we obtain

$$-\frac{1}{2i\pi}\int_{i\mathbb{R}} \Psi_n(s+t+z,-z)\frac{dz}{z+y_1+y_2} .$$

Repeating the process for the three other terms, we get:

$$\begin{aligned} A_n(s,t) = \lim_{z_1,z_2\to 0,\Re(z_1),\Re(z_2)>0} &(\frac{1}{2i\pi})^2 \int_{i\mathbb{R}}\int_{i\mathbb{R}} \frac{1}{2i\pi}\int_{i\mathbb{R}} \Big\{ \\ &\sigma^*_{11}(s+y_1)\sigma^*_{21}(-y_1)\sigma^*_{12}(s+y_2)\sigma^*_{22}(-y_2)\frac{1}{z+y_1+y_2} \\ &+\sigma^*_{11}(s+y_1)\sigma^*_{21}(-y_1)\sigma^*_{12}(-y_2)\sigma^*_{22}(s+y_2)\frac{1}{z+y_1-y_2-t} \\ &+\sigma^*_{11}(-y_1)\sigma^*_{21}(s+y_1)\sigma^*_{12}(s+y_2)\sigma^*_{22}(-y_2)\frac{1}{z+y_2-y_1-s} \\ &+\sigma^*_{11}(-y_1)\sigma^*_{21}(s+y_1)\sigma^*_{12}(-y_2)\sigma^*_{22}(s+y_2)\frac{1}{z-y_1-y_2-s-t}\Big\} \\ &\Psi_n(s+t+z,-z)dz\frac{dy_1}{y_1-z_1}\frac{dy_2}{y_2-z_2} . \end{aligned} \tag{20}$$

Likewise, the function $B_n(s,t)$ has a similar form. This proves the theorem.

4 Application to Exponential Distributions

Assume now that the four input sequences are made of i.i.d. exponential variables, and let a, b, c and d be the parameters of the variables $\sigma_{11}, \sigma_{12}, \sigma_{21}$ and σ_{22} respectively. This amounts to say that:

$$S(x,y,u,v) = (1-e^{a(u-x)})(1-e^{b(u-y)})(1-e^{c(v-x)})(1-e^{d(v-y)}).$$

If $a = d$ and $b = c$, then it is easy to convince oneself that the variables $X_1(n)$ and $X_2(n)$ are exchangeable for any n, that is to say the function $F_n(x,y)$ is symmetric in (x,y), and the function $\Psi_n(s,t)$ is symmetric in (s,t). In that case, $A(s,t) = B(t,s)$. We shall call this case "semi-symmetric" hereafter.

Instead of applying the general framework developed in the previous section, we adopt a more direct approach to obtain the functional recurrence on Ψ_n. Indeed, it is simpler to obtain the kernel K of theorem 3.1 by integrating first equation (19) than by using formula (20).

In this semi-symmetric case, we have from (19):

$$\int_{\mathbb{R}}\int_{\mathbb{R}_\infty} S(x+u,y+u,u,v)e^{-sx-ty}dxdy$$
$$= \int_0^\infty (1-e^{-ax})(1-e^{b(v-u-x)})e^{-sx}dx \int_0^\infty (1-e^{-by})(1-e^{a(v-u-y)})e^{-ty}dy$$
$$= \left(\frac{a}{s(s+a)} - \frac{ae^{b(v-u)}}{(s+b)(s+a+b)}\right)\left(\frac{b}{t(t+b)} - \frac{be^{a(v-u)}}{(t+a)(t+a+b)}\right).$$

The computation of Ψ_{n+1} is therefore reduced to that of:

$$A_n(s,t) = 2ab\int_0^\infty \int_0^u e^{-u(s+t)}\left(\frac{1}{s+a} - \frac{se^{b(v-u)}}{(s+b)(s+a+b)}\right) \left(\frac{1}{t+b} - \frac{te^{a(v-u)}}{(t+a)(t+a+b)}\right) dF_n(u,v)\ . \tag{21}$$

4.1 The totally symmetric case

We assume in this section that $a = b$. We will actually take $a = 1$. It can be easily checked that $A(s,t)$ is now symmetric in (s,t) so that the expression for Ψ_{n+1} reduces even more to yield:

$$\Psi_{n+1}(s,t)$$
$$= \frac{2}{(s+1)(t+1)}\int_0^\infty \int_0^u e^{-u(s+t)}\left(1-\frac{se^{v-u}}{s+2}\right)\left(1-\frac{te^{v-u}}{t+2}\right)dF_n(u,v)$$
$$= \frac{2}{(s+1)(t+1)}\Big(J_n(s+t,0) - \big(\frac{s}{s+2}+\frac{t}{t+2}\big)J_n(s+t+1,-1)$$
$$+\frac{st}{(s+2)(t+2)}J_n(s+t+2,-2)\Big)\ ,$$

where

$$J_n(s,t) = \int_{\mathbb{R}}\int_0^x e^{-sx-ty}dF_n(x,y)\ . \tag{22}$$

In order to obtain a functional recurrence for the sequence Ψ_n, there remains to express the function J_n in terms of Ψ_n. This can miraculously be done using lemma A.2 in appendix A. One obtains (the function K_n is null if $n > 0$):

$$J_n(s,t) = -\frac{1}{2i\pi}\int_{i\mathbb{R}} \Psi_n(s+t+z,-z)\frac{dz}{z+t}\ ,$$

for $\Re(t) < 0$. This leads to:

$$\Psi_{n+1}(s,t) = -\frac{2}{(s+1)(t+1)(s+2)(t+2)}$$
$$\frac{1}{2i\pi}\int_{i\mathbb{R}}\left((s+2)(t+2)\frac{1}{z} - 2(st+s+t)\frac{1}{z-1} + st\frac{1}{z-2}\right) \tag{23}$$
$$\Psi_n(s+t+z,-z)dz\ .$$

In this integral, the term "$1/z$" is to be interpreted as: "$\lim_{\varepsilon\to 0+} 1/(1-\varepsilon)$". This follows from the fact that for any fixed $x \in i\mathbb{R}$, the function $\varepsilon \mapsto J_n(x+\varepsilon,-\varepsilon)$ is analytic for $\Re(\varepsilon) > 0$ (see the Appendix), so that $J_n(x,0)$ can be obtained from the integral formula above by letting t go to 0^-.

When $n = 0$, we substitute $\Psi_0(s,t) = 1$ in this equation, which becomes:

$$\Psi_1(s,t)$$
$$= \frac{2}{(s+1)(t+1)(s+2)(t+2)} \frac{1}{2} ((s+2)(t+2) - 2(st+s+t) + st) \quad (24)$$
$$= \frac{4}{(s+1)(t+1)(s+2)(t+2)} ,$$

by virtue of lemma A.5. This computation was of course hardly necessary: one obtains directly from (16) the value $F_1(x,y) = (1-e^{-x})^2(1-e^{-y})^2$.

The computation of Ψ_2 is much more interesting, as it gives the intuition for the general result. We shall omit it for the sake of brevity, and state instead the principal lemma.

LEMMA 4.1 *For all $n \geq 1$, the function Ψ_n has the following form:*

$$\Psi_n(s,t) = \frac{4^n}{(s+1)(s+2)(t+1)(t+2)} \frac{\mathcal{N}(s,t)}{\mathcal{D}(s+t)} ,$$

where:

$$\mathcal{D}(x) = [(x+1)(x+2)^2(x+3)^2(x+4)^2]^{n-1}$$
$$\mathcal{N}(s,t) = P_n(s+t) + st\, Q_n(s+t) ,$$

and P_n, Q_n are polynomials. The degree of P_n is $3(n-1)$, and the degree of Q_n is $3n-5$ for $n > 1$.

PROOF The proof is by induction. The case $n = 1$ is already solved in view of (24): take $P_1(x) = 1$ and $Q_1(x) = 0$. The degree of P_1 is zero.

Assume the lemma holds for some n. Then, by (23), we have, denoting $x = s + t$:

$$\Psi_{n+1}(s,t) = -\frac{2}{(s+1)(t+1)(s+2)(t+2)}$$
$$\frac{1}{2i\pi} \int_{i\mathbb{R}} \left((s+2)(t+2)\frac{1}{z} - 2(st+x)\frac{1}{z-1} + st\frac{1}{z-2} \right)$$
$$\frac{P_n(x) - z(x+z)Q_n(x)}{\mathcal{D}_n(x)(x+z+1)(x+z+2)(z-1)(z-2)} dz . \quad (25)$$

All that is needed is therefore a way to compute the integral:

$$I(\zeta) = \frac{1}{2i\pi} \int_{i\mathbb{R}} \frac{P_n(x) - z(x+z)Q_n(x)}{(x+z+1)(x+z+2)(z-1)(z-2)} \frac{dz}{z-\zeta}$$

for a fixed, pure imaginary number x and a nonnegative number ζ (see the remark above concerning the term $1/z$).

The integrand of $I(\zeta)$ is a rational function in z, and the degree of the numerator (=2) is less than that of the denominator (=5). A decomposition in elementary fractions yields:

$$\frac{P_n(x) - z(x+z)Q_n(x)}{(x+z+1)(x+z+2)(z-1)(z-2)(z-\zeta)} =$$
$$\frac{\alpha(x,\zeta)}{x+z+1} + \frac{\beta(x,\zeta)}{x+z+2} + \frac{\gamma(x,\zeta)}{z-1} + \frac{\delta(x,\zeta)}{z-2} + \frac{\varepsilon(x,\zeta)}{z-\zeta} . \quad (26)$$

With the use of lemma A.5, integration of this equation yields:

$$\begin{aligned} I(\zeta) &= \alpha(x,\zeta)sgn(-1-x)+\beta(x,\zeta)sgn(-2-x)+\gamma(x,\zeta)sgn(1) \\ &\quad +\delta(x,\zeta)sgn(2)+\varepsilon(x,\zeta)sgn(\zeta) = \frac{1}{2}(\alpha+\beta-\gamma-\delta-\varepsilon)\ . \end{aligned}$$

On the other hand, it is clear from the decomposition (26) that $\alpha+\beta+\gamma+\delta+\varepsilon = 0$ (multiply by z and let z go to infinity). Therefore, $I(\zeta) = \alpha(x,\zeta)+\beta(x,\zeta)$. The values of $\alpha(x,\zeta)$ and $\beta(x,\zeta)$ are respectively:

$$\begin{aligned} \alpha(x,\zeta) &= -\frac{P_n(x)-(x+1)Q_n(x)}{(x+2)(x+3)(x+1+\zeta)} \\ \beta(x,\zeta) &= -\frac{P_n(x)-2(x+2)Q_n(x)}{(x+3)(x+4)(x+2+\zeta)}\ . \end{aligned}$$

Plugging these expressions in (25) and reducing to the same denominator (the help of the MAPLE software was valuable here), one obtains:

$$\begin{aligned} \Psi_{n+1}(s,t) &= -\frac{2}{(s+1)(t+1)(s+2)(t+2)} \\ &\frac{2}{D_n(x)}\frac{A(x)P_n(x)+B(x)Q_n(x)+st(C(x)P_n(x)+D(x)Q_n(x))}{(x+1)(x+2)^2(x+3)^2(x+4)^2}\ , \end{aligned} \tag{27}$$

where:

$$\begin{aligned} A(x) &= 2(x+4)(5x^2+20x+18) \\ B(x) &= x(x+1)^2(x+4)(3x+8) \\ C(x) &= 5x+14 \\ D(x) &= (x+1)(x^2-8)\ . \end{aligned}$$

The lemma is therefore proved, and the polynomials P_{n+1} and Q_{n+1} are given by the recurrence:

$$\begin{pmatrix} P_{n+1}(x) \\ Q_{n+1}(x) \end{pmatrix} = \begin{pmatrix} A(x) & B(x) \\ C(x) & D(x) \end{pmatrix} \begin{pmatrix} P_n(x) \\ Q_n(x) \end{pmatrix}\ . \tag{28}$$

One easily checks that the degree of P_{n+1} is $\max\{3+3(n-1), 4+3n-5\} = 3n$ and that of Q_{n+1} is 2 if $n=1$ and $\max\{3(n-1)+1, 3n-5+3\} = 3n-2$ if $n>1$.

The recurrence (28) may be solved by elementary means.

LEMMA 4.2 *For all $n \geq 1$, the polynomials P_n and Q_n are given by:*

$$\begin{aligned} P_n(x) &= \frac{1}{\lambda_1(x)-\lambda_2(x)}\left[\lambda_1^n(x)-\lambda_2^n(x)-D(x)\left(\lambda_1^{n-1}(x)-\lambda_2^{n-1}(x)\right)\right] \\ Q_n(x) &= \frac{-C(x)}{\lambda_1(x)-\lambda_2(x)}\left[\lambda_1^{n-1}(x)-\lambda_2^{n-1}(x)\right](*)\ , \end{aligned}$$

where

$$\lambda_i(x) = \frac{1}{2}\left(A(x) + D(x) \pm \sqrt{(A(x) - D(x))^2 + 4B(x)C(x)}\right) .$$

PROOF Let $Y_n(x)$ be the column vector whose entries are $P_n(x)$ and $Q_n(x)$, and let $M(x)$ denote the 2×2 matrix which appears in (28). We then have: $Y_{n+1}(x) = M(x)Y_n(x)$ for $n \geq 1$, with $Y_1(x) = (1,0)^t$. If $\mathcal{Y}(z,x) = \sum_{n=1}^{\infty} Y_n(x)z^{n-1}$, then $\mathcal{Y}$ obeys the matrix equation: $\mathcal{Y}(z,x)(I - zM(x)) = Y_1(x)$. Inversion of this equation (at points where $I - zM(x)$ is not singular) yields:

$$\mathcal{Y}(z,x) = \frac{1}{P(z,x)} \begin{pmatrix} 1 - zD(x) & -zB(x) \\ -zC(x) & 1 - zA(x) \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} , \tag{29}$$

where $P(z,x) = det(I - zM(x)) = z^2(A(x)D(x) - B(x)C(x)) - z(A(x) + D(x)) + 1 = (1 - z\lambda_1(x))(1 - z\lambda_2(x))$. An expansion in series of z of the right-hand side of (29), followed by the identification of the coefficients proves the lemma. Note that such computations are known as the *spectral expansion* in the literature.

We are now in position to state a number of properties of the solution of the system (1):

PROPOSITION 4.3 *Let $\{(X_1(n), X_2(n))\}_0^{\infty}$ be the solution of (1). We have:*

$$EX_1(n) = EX_2(n) = \frac{407}{228}\, n + \frac{344}{1083} + \frac{126}{361}\frac{(-1)^n}{18^n} . \tag{30}$$

In particular, the Lyapunov exponent of the system is:

$$\gamma = \lim_{n \to \infty} \frac{X_1(n)}{n} = \frac{407}{228} . \tag{31}$$

Let $\Delta(n) = X_1(n) - X_2(n)$. Then:

$$\begin{aligned} P(\Delta(n) \leq x) &= \frac{1}{19}\left(\frac{23}{2}e^x - 2e^{2x} + \frac{7}{6}\frac{(-1)^{n-1}}{18^{n-1}}(e^x - e^{2x})\right)\mathbf{1}_{\{x<0\}} \\ &\quad + \frac{1}{19}\left(19 - \frac{23}{2}e^{-x} + 2e^{-2x} - \frac{7}{6}\frac{(-1)^{n-1}}{18^{n-1}}(e^{-x} - e^{-2x})\right)\mathbf{1}_{\{x \geq 0\}} . \end{aligned}$$

In particular, $\Delta(n)$ converges (geometrically fast) in law to a RV $\Delta(\infty)$ with distribution ([10]):

$$\begin{aligned} P(\Delta(\infty) \leq x) &= \frac{1}{19}\left(\frac{23}{2}e^x - 2e^{2x}\right)\mathbf{1}_{\{x<0\}} \\ &\quad + \frac{1}{19}\left(19 - \frac{23}{2}e^{-x} + 2e^{-2x}\right)\mathbf{1}_{\{x \geq 0\}} . \end{aligned}$$

PROOF The Laplace transform of $X_1(n)$ is:

$$\begin{aligned} X_n^*(s) = \Psi_n(s,0) &= \frac{4^n}{2(s+1)(s+2)} \frac{\mathcal{N}(s,0)}{\mathcal{D}(s)} \\ &= \frac{4^n}{2(s+1)(s+2)} \frac{P_n(s)}{[(s+1)(s+2)^2(s+3)^2(s+4)^2]^{n-1}} . \end{aligned}$$

Formula (30) follows by differentiation, with or without the help of MAPLE.

Likewise, the Laplace transform of $\Delta(n)$ is given by:

$$\begin{aligned}\Delta_n^*(s) = \Psi_n(s,-s) &= \frac{4^n}{(1+s)(2+s)(1-s)(2-s)} \frac{P_n(0) - s^2 Q_n(0)}{\mathcal{D}(0)} \\ &= \frac{1}{19} \frac{76 - 7s^2(1-(-18)^{1-n})}{(1+s)(2+s)(1-s)(2-s)} .\end{aligned}$$

Inverting this Laplace transform yields the distribution of $\Delta(n)$.

Other properties of the couple $(X_1(n), X_2(n))$ can be derived from the value of Ψ_n, such as their higher moments and their covariance.

4.2 The semi-symmetric case

We now turn to the solution of the semi-symmetric case. Il follows the same steps as in the totally symmetric case, and many details will be omitted. It should be emphasized here that the use of an algebraic manipulation package (in our case, MAPLE) seems essential to be able to complete these calculations.

At the end of section 3, we were left with the formula: $\Psi_{n+1}(s,t) = st(A_n(s,t) + A_n(t,s))$, A_n being given by (21). Expanding the product gives:

$$A_n(s,t) = \tag{32}$$

$$\begin{array}{ll} ab[(s+a)(s+b)(t+a)(t+b)(s+a+b)\ (t+a+b)]^{-1} & \\ (\ (s+b)(t+a)(s+a+b)(t+a+b) & J_n(s+t,0) \\ -t(s+b)(t+b)(s+a+b) & J_n(s+t+a,-a) \\ -s(s+a)(t+a)(t+a+b) & J_n(s+t+b,-b) \\ +(s+a)(t+b) & J_n(s+t+a+b,-a-b)\ ) \ , \end{array}$$

where J_n is still given by (22). As in section 4.1, the computation of Ψ_2 allows to guess what the general structure of Ψ_n is. Again, we skip it to state directly:

LEMMA 4.4 *For all $n \geq 1$, the function Ψ_n has the following form:*

$$\Psi_n(s,t) = \frac{(ab)^{n+1}}{(s+a)(s+b)(s+a+b)(t+a)(t+b)(t+a+b)} \frac{\mathcal{N}_n(s,t)}{\mathcal{D}(s+t)^{n-1}} ,$$

where:

$$\begin{aligned}\mathcal{D}(x) &= (a+x)(b+x)(a+b+x)^2(2a+x)^2(2b+x)^2 \\ &\qquad (2b+a+x)^2(2a+b+x)^2(2a+2b+x)^2 \\ \mathcal{N}_n(s,t) &= P_n(s+t) + st\, Q_n(s+t) + (st)^2\, R_n(s+t),\end{aligned}$$

and P_n, Q_n and R_n are polynomials.

PROOF The proof is similar to that of lemma 4.1. The case $n = 1$ is easily proved since it is clear that $F_1(x,y) = (1-e^{-ax})(1-e^{-bx})(1-e^{-ay})(1-e^{-by})$,

the Laplace Transform of which is $\Psi_1(s,t) = a^2b^2(a+b+2s)(a+b+2t)/(s+a)(s+b)(s+a+b)(t+a)(t+b)(t+a+b)$. We have therefore $P_1(x) = 2x(a+b)+(a+b)^2$, $Q_1(x) = 4$ and $R_1 = 0$.

For the general case, one computes the integral

$$I(\zeta) = \frac{1}{2i\pi}$$

$$\int_{i\mathbb{R}} \frac{P_n(x) - z(x+z)Q_n(x) + z^2(x+z)^2R_n(x)}{(x+z+a)(x+z+b)(x+z+a+b)(a-z)(b-z)(a+b-z)} \frac{dz}{z-\zeta}$$

by expanding the integrand in elementary fractions of z (the degree of the numerator is still less than that of the denominator). The use of the induction hypothesis and algebraic manipulations show that the vector of polynomials $Y_n = (P_n(x), Q_n(x), R_n(x))^t$ is given by a linear recurrence $Y_{n+1} = M(x)Y_n(x)$, as in (28). A MAPLE program implementing this recurrence and computing the Lyapunov exponent of the system is given in appendix 5.3. □

Repeating the analysis of the totally symmetric case, we introduce the vector $Y_n(x) = (P_n(x),\ Q_n(x),\ R_n(x))^t$ and the generating function $\mathcal{Y}(z,x) = \sum_{n=1}^{\infty} Y_n(x)z^{n-1}$, which satisfies:

$$\mathcal{Y}(z,x) = \frac{[I - zM(x)\tilde{]}Y_1(x)}{(1-z\lambda_1(x))(1-z\lambda_2(x))(1-z\lambda_3(x))}, \tag{33}$$

where $[A\tilde{]}$ denotes the comatrix of some matrix A, and $(1-z\lambda_1(x))(1-z\lambda_2(x))(1-z\lambda_3(x))$ is the determinant of $I-zM(x)$. It follows that $Y_n(x)$ can be expressed in terms of the function

$$\phi_n(x) = [z^{n-1}]\frac{1}{(1-z\lambda_1(x))(1-z\lambda_2(x))(1-z\lambda_3(x))} = \tag{34}$$

$$\frac{\lambda_1(x)^{n+1}(\lambda_3(x)-\lambda_2(x)) + \lambda_2(x)^{n+1}(\lambda_1(x)-\lambda_3(x)) + \lambda_3(x)^{n+1}(\lambda_2(x)-\lambda_1(x))}{(\lambda_1(x)-\lambda_2(x))(\lambda_2(x)-\lambda_3(x))(\lambda_3(x)-\lambda_1(x))},$$

at points where all three roots $\lambda_i(x)$ are distinct. A similar expansion exists in the other cases. We skip the painful details as the explicit expansion of P_n, Q_n and R_n (and therefore Ψ_n) is too complicated to learn us anything. Note however that if the parameters a and b are given a particular numerical value, computations with MAPLE become again feasible, and the results are compact enough.

We instead concentrate on the computation of the Lyapunov exponent of the system. This is a simpler problem owing to the fact that the Laplace transform of $X_1(n)$ is simply:

$$X_1^*(n)(s) = \frac{(ab)^n}{(a+b)(a+s)(b+s)(a+b+s)} \frac{1}{(a+s)^{n-1}(b+s)^{n-1}} \frac{P_n(s)}{[\mathcal{D}(x)]^{n-1}},$$

where $\mathcal{D}$ is given in Lemma 4.4. The expected value of $X_1(n)$ is therefore:

$$EX_1(n) = -\frac{P_n'(0)}{P_n(0)} + (n-1)$$

$$\left(\frac{1}{a}+\frac{1}{b}+2\left(\frac{1}{a+b}+\frac{1}{2a}+\frac{1}{2b}+\frac{1}{2b+a}+\frac{1}{2a+b}+\frac{1}{2(a+b)}\right)\right)$$

$$+\frac{1}{a}+\frac{1}{b}+\frac{1}{a+b}\ . \tag{35}$$

The value of $P_n(0)$ is easily shown to be $P_1(0)\lambda_1(0)^{n-1}$, where $\lambda_1(0) = M_{1,1}(0) = 64a^2b^2(a+2b)^2(2a+b)^2(a+b)^4$: this is simply due to the fact that $M_{1,2}(0) = M_{1,3}(0) = 0$. This eigenvalue is actually the *dominant* eigenvalue of $M(0)$, that is, the one with largest modulus (which is not shown here).

Let $e_1 = (1,0,0)$. Then we have:

$$e_1\mathcal{Y}(z,0) = \sum_{n=1}^{\infty} z^{n-1}P_n(0) = \frac{P_1(0)}{1-\lambda_1(0)z} =$$

$$\frac{e_1[I - zM(0)\tilde{]}Y_1(0)}{(1-z\lambda_1(0))(1-z\lambda_2(0))(1-z\lambda_3(0))}$$

so that:

$$e_1[I - zM(0)\tilde{]}Y_1(0) = P_1(0)(1-z\lambda_2(0))(1-z\lambda_3(0))\ . \tag{36}$$

Differentiating equation (33) with respect to x, we have:

$$\sum_{n=1}^{\infty} P_n'(0)z^{n-1} = \frac{\partial\mathcal{Y}}{\partial x}(z,0)$$

$$= \frac{z\lambda_1'(0)\ e_1[I - zM(0)\tilde{]}Y_1(0)}{(1-z\lambda_1(0))^2(1-z\lambda_2(0))(1-z\lambda_3(0))} + O\left(1/(1-z\lambda_1(0))\right)\ .$$

The right-hand side of this formula is a rational function of z whose first singularity is at $z = 1/\lambda_1(0)$ (because $\lambda_1(0)$ is the dominant eigenvalue of $M(0)$). This singularity is a pole of order 2. It follows (see for instance [6]) that:

$$P_n'(0) = [z^{n-1}]\frac{\partial\mathcal{Y}}{\partial x}(z,0) =$$

$$(n-1)\lambda_1(0)^{n-2}\frac{\lambda_1'(0)\ e_1[I - (\lambda_1(0))^{-1}M(0)\tilde{]}Y_1(0)}{(1-\lambda_2(0)/\lambda_1(0))(1-\lambda_3(0)/\lambda_1(0))}\ (1+o(1))\ .$$

Using (36), we obtain:

$$\frac{P_n'(0)}{P_n(0)} = n\ \frac{\lambda_1'(0)}{\lambda_1(0)}\ (1+o(1))\ .$$

It remains to evaluate $\lambda_1'(0)$, which we do as in theorem 2.2. If $\Pi(z,x) = (z-\lambda_1(x))(z-\lambda_2(x))(z-\lambda_3(x))$ is the characteristic polynomial of $M(x)$, we have:

$$\lambda_1'(0) = -\left.\frac{\partial \Pi(z,x)}{\partial x}\right|_{z=\lambda_1(0),x=0} \left(\left.\frac{\partial \Pi(z,x)}{\partial z}\right|_{z=\lambda_1(0),x=0}\right)^{-1}. \tag{37}$$

In order that these computations be perfectly valid, it remains to be proved that the eigenvalue $\lambda_1(x)$ is indeed the dominant one for all values of x. A way to obtain this result would be to show that all three $\lambda_i(x)$, $i = 1, 2, 3$ are never equal in modulus (actually, all seem to be real). As $\lambda_1(x)$ is dominant at $x = 0$ (which is also to be proved), it is then dominant everywhere.

A MAPLE program based on equations (35) and (37) is given in appendix B. It produces (among other things) the following result:

LEMMA 4.5 *Let $r = a/b$. The Lyapunov exponent of the system is given by:*

$$\gamma = \frac{1}{16a(a+1)} \frac{N(r)}{D(r)},$$

where

$$\begin{aligned} N(r) &= 160r^{10} + 1776r^9 + 8220r^8 + 21378r^7 + 35595r^6 + 41566r^5 \\ &\quad +35595r^4 + 21378r^3 + 8220r^2 + 1776r + 160 \\ D(r) &= 8r^8 + 80r^7 + 321r^6 + 690r^5 + 880r^4 + 690r^3 + 321r^2 + 80r + 8 \,. \end{aligned}$$

A plot of this function is given in figure 2 (for $a = 1$). Note that $D(r)$ and $N(r)/r$ are symmetric polynomials in r, which reflects the fact that γ is symmetric in (a, b). The limit of γ when $r \to \infty$ is $5/4$. The case $a = \infty$ actually admits a simpler solution, which we detail at the end of this section.

To conclude the semi-symmetric case, we give a description of the behavior of the RV $\Delta(n)$:

PROPOSITION 4.6 *The distribution of $\Delta(n)$ converges geometrically fast to that of $\Delta(\infty)$.*

PROOF The Laplace transform of $\Delta(n)$ is:

$$\begin{aligned} &\Delta^*(n)(u) = \Psi_n(u, -u) \\ &= \frac{a^2b^2(a+b)^2}{(a^2-u^2)(b^2-u^2)(a+b)^2 - u^2)} \frac{P_n(0) - u^2Q_n(0) + u^2R_n(0)}{(ab)^{n-1}\lambda_1(0)} \\ &= \frac{a^2b^2(a+b)^2}{(a^2-u^2)(b^2-u^2)(a+b)^2 - u^2)} \left(1 - u^2\frac{Q_n(0)}{P_n(0)} + u^4\frac{R_n(0)}{P_n(0)}\right) . \end{aligned}$$

With the expression of $P_n(0), Q_n(0)$ and $R_n(0)$ in terms of the function ϕ_n (eq. (34)), and knowing that $\lambda_1(0)$ is the dominant eigenvalue, one obtains that the last term satisfies:

$$\begin{aligned} (\ldots) &= 1 - u^2\frac{\lambda_1(0)\left[Q_2(0) - (\lambda_2(0) + \lambda_3(0))Q_1(0) - u^2R_2(0)\right]}{P_1(0)(\lambda_1(0) - \lambda_2(0))(\lambda_1(0) - \lambda_3(0))} \\ &\quad + O(\rho^n) \,, \end{aligned}$$

where $\rho = \max\{|\lambda_2(0)|, |\lambda_3(0)|\}/|\lambda_1(0)$. This proves the theorem. The limiting distribution can be computed from the expression above. It does not seem to possess a simple expression, so we skip it. □

The case $a = b$

When $a = b$, the results of lemma 4.4 should reduce to that of lemma 4.1. This is not obvious at first sight, and some algebra is involved to check this reduction. It turns out that when $a = b$, the 3×3 matrix $M(x)$ involved in the computation of the polynomials P_n, Q_n and R_n has a rank equal to 2 for all x. This explains why the numerator $\mathcal{N}_n$ of lemma 4.4 can be generated with a linear recurrence involving 2×2 matrices. One checks that this matrix is indeed the matrix of (28).

The case $a = \infty$

This corresponds to the interesting case where, in a two-processor system, computation times are negligible compared to interprocessor communication delays.

PROPOSITION 4.7 *For any $n \geq 1$, the Laplace transform of the couple $(X_1(n), X_2(n))$ is given by:*

$$\Psi_n(s,t) = \frac{b^{2n}}{(b+s)(b+t)} \left(\frac{3(s+t)+4b}{(s+t+b)(s+t+2b)^2} \right)^{n-1} .$$

One has:

$$EX_1(n) = EX_2(n) = \frac{5n-1}{4b} .$$

The Lyapunov exponent of the system is $5/4$. For any $n \geq 1$, the distribution of $\Delta(n)$ is independent of n and is:

$$P(\Delta(n) \leq x) = 1/2\, e^{-bx} \mathbf{1}_{\{x<0\}} + (1 - e^{-bx})/2\, \mathbf{1}_{\{x \geq 0\}} .$$

PROOF

It is easy to see that the recurrence on Ψ_n is now reduced to:

$$\Psi_{n+1}(s,t) =$$

$$\left(\frac{b}{b+s} + \frac{b}{b+t} \right) J_n(s+t,0) - \frac{b(s+t)}{(b+s)(b+t)} J_n(s+t+b,-b)$$

$$= \frac{b}{(b+s)(b+t)} \left((2b+s+t) J_n(s+t,0) - (s+t) J_n(s+t+b,-b) \right) .$$

It then follows that Ψ_n is of the form: $\Psi_n(s,t) = b^{2n} P_n(s+t)/[(b+s+t)(2b+s+t)^2]^{n-1}$, and that $P_{n+1}(x) = (3x+4b)P_n(x)$. The result follows. □

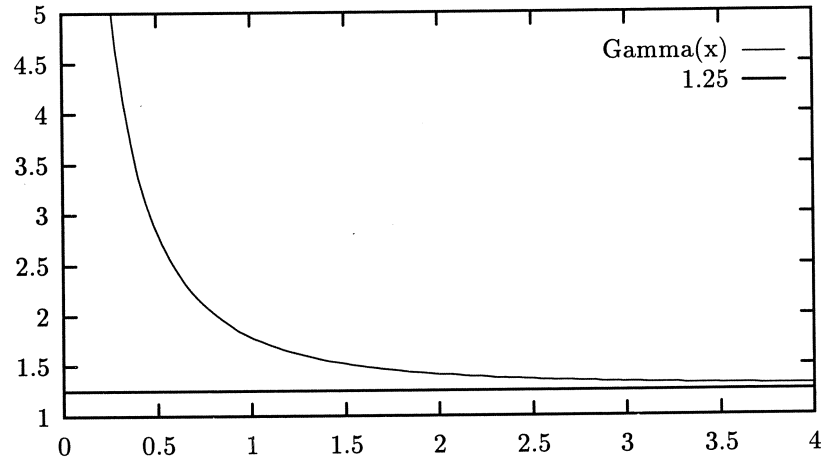


FIGURE 2. The curve $r \mapsto \gamma$ (*)

4.3 Extensions

We indicate a number of possibilities to extend the computations to other models/assumptions.

- General Exponential case. The symmetry between A_n and B_n disappears, and $\Psi_n(s,t)$ is not symmetric anymore in (s,t). However, it is still a rational function, and it is likely that a polynomial recurrence may be used, based on the decomposition of the numerator of Ψ_n isolating s and t (as in the general Bernoulli case, see section 5).

- Other distributions. The possibility to achieve the computations is strongly connected to the relative simplicity of the inner integral that appears in the definition of A_n. In the (totally symmetric) Erlang-2 case, polynomial terms appear, which lead to an expression of Ψ_{n+1} in terms of the function J_n *and* of its derivatives. However, the derivatives of J_n can be computed from the very formulas of Appendix A without further complications.

- Higher dimensions. It is of course very limitating to be able to analyze only two dimensional systems. In principle, the technique described here may be applied to systems with more than two state variables, through a quite straightforward induction. However, the iterations of the formulas of Appendix A to compute functions like $E(exp(-s_1X_1 - s_2X_2 -$

$\ldots s_n X_n)\mathbf{1}\{X_1 > X_2 > \ldots X_n\}$) becomes quickly problematic. A lot of guessing is involved in order to obtain the general form of the solution.

- (min, +)-systems. The theory above does not assume *a priori* that random variables should be positive. Hence, applying the technique to negative RV's and changing signs, one may compute the performance of simple linear systems in the $(min, +)$ dioid.

- Other dynamical systems. The approach of section 2 has been applied successfully to dynamic systems described by evolution equations contain the min, max and + operators [9]. It actually turns out that the Markov chain technique applies to every system with an evolution property satisfying a certain time-homogeneity condition (called "1-linearity" in [12]).

- Other models with synchronizations. The technique developed above seems to allow the analysis (both in transient and stationary regime) of some basic two-node queueing systems with synchronization whose evolution equations involve only addition and maximization (or minimization, *cf.* a previous remark). Such systems include the Fork/Join queue, with evolution equations (on the waiting time):

$$W_n^1 = [W_{n-1}^1 + \sigma_{n-1}^1 - \tau_n]^+$$
$$W_n^2 = [W_{n-1}^2 + \sigma_{n-1}^2 - \tau_n]^+,$$

(solved in [7] in steady state with complicated computations) and the round-robin routing model, with evolution equations (for the workload at instants of arrival in the system):

$$W_n^1 = [W_{n-1}^1 + u_n^1 \sigma_{n-1} - \tau_n]^+$$
$$W_n^2 = [W_{n-1}^2 + u_n^2 \sigma_{n-1} - \tau_n]^+,$$

where $u_n^1 = 1$ iff n is odd, and $u_n^1 + u_n^2 = 1$. For the last model, qualitative results are known (mainly, stochastic comparisons with the Bernoulli routing scheme) but hardly any quantitative ones. Moreover, no results are known if resequencing is added before exiting the system.

5 Return to the Discrete Case

The case of lattice distributions can be handled in principle with the approach of the previous section. There are however a few particularities in the analysis of discrete distributions. The analysis is similar to the previous one, with Laplace transforms replaced by generating functions, and the contour $i\mathbb{R}$ replaced by the unit circle (see Appendix A).

Section 2 described how the use of a Markov chain underlying the evolution of the system allows the solution of the problem by linear algebra techniques. However, if the support of the distributions involved is infinite, the matrices

involved are also infinite, and the practical solution of the problem is not evident. The approach which follows allows to handle more difficult cases, such as geometric distributions (section 5.3). In sections 5.1 and 5.2, we apply it first to a case of finite (Bernoulli) distributions.

As before, let $F_n(x,y) = \mathbb{P}\{X_1(n) \le x, X_2(n) \le y\}$, and let now $f_n(x,y) = \mathbb{P}\{X_1(n) = x, X_2(n) = y\}$. Set: $\Phi_n(s,t) = \mathbb{E}(s^{X_1(n)}t^{X_2(n)})$. This generating function is defined (at least) on the set $\{|s| = |t| = 1\}$. We have:

$$\sum_x \sum_y s^x t^y F_n(x,y) = \frac{1}{1-s}\,\frac{1}{1-t}\,\Phi_n(s,t)\ .$$

The recurrence on the distributions, viz.:

$$F_{n+1}(x,y) \;=\; \int_{-\infty}^{\inf(x,y)} \int_{-\infty}^{\inf(x,y)} S(x,y,u,v)dF_n(u,v)\ ,$$

translates now into:

$$\frac{1}{1-s}\frac{1}{1-t}\Phi_{n+1}(s,t) \;=\; A_n(s,t) + B_n(s,t) + C_n(s,t)\ ,$$

with:

$$\begin{aligned}
A_n(s,t) &= \sum\sum (st)^u \mathbf{1}_{\{u>v\}} \left(\textstyle\sum_{x,y\ge 0} S(x+u,y+u,u,v)s^x t^y\right) f_n(u,v)\\
B_n(s,t) &= \sum\sum (st)^u \mathbf{1}_{\{u=v\}} \left(\textstyle\sum_{x,y\ge 0} S(x+u,y+u,u,u)s^x t^y\right) f_n(u,v)\\
C_n(s,t) &= \sum\sum (st)^v \mathbf{1}_{\{u<v\}} \left(\textstyle\sum_{x,y\ge 0} S(x+v,y+v,u,v)s^x t^y\right) f_n(u,v)\ ,
\end{aligned} \tag{38}$$

and

$$S(x,y,u,v) \;=\; \mathbb{P}(\sigma_{11} \le x-u, \sigma_{12} \le y-u, \sigma_{21} \le x-v, \sigma_{22} \le y-v)\ .$$

According to the results of section 3, and using lemma A.4, it is possible to prove that the functions A_n, B_n and C_n are computed from Φ_n *via* contour integrals involving some kernel. On the other hand, the explicit evaluation of the inner sums often provides a more direct way to actually construct the recurrence on Φ_n.

5.1 The totally symmetric Bernoulli case

In this section, it is assumed that the four input sequences are made of Bernoulli variables with parameter $a = P(\sigma = 1)$. For convenience, let $\bar{a} = 1 - a$.

PROPOSITION 5.1 *The Lyapunov exponent of the system is given by:*

$$\gamma(a) \;=\; \frac{a(a-2)(2a^3-a^2-2a+2)}{1+2a(a-1)(a^2-3a+1)} \;=\; 1 - \frac{(2a+1)(1-a)^4}{1+2a(a-1)(a^2-3a+1)}\ . \tag{39}$$

In particular, $\gamma(1/2) = 6/7$ *([10]). The random variable* $\Delta(n)$ *converges exponentially fast to a* $\Delta(\infty)$ *given by:*

$$\mathbb{P}(\Delta(\infty)=0)=\frac{1-2a\bar{a}}{1+2a\bar{a}(a^2-3a+1)},$$

$$\mathbb{P}(\Delta(\infty)=\pm 1)=\frac{a(2-a)\bar{a}^2}{1+2a\bar{a}(a^2-3a+1)}.$$

The rate of convergence is given by:

$$\epsilon(a) \;=\; 2a(a-1)(a^2-3a+1)\;.$$

PROOF In this case, we have:

$$\sum_{x,y\geq 0} S(x+u,y+u,u,v)s^x t^y$$

$$=\left((1-a)(1-a+a\mathbf{1}_{\{u>v\}})+\frac{s}{1-s}\right)\left((1-a)(1-a+a\mathbf{1}_{\{u>v\}})+\frac{t}{1-t}\right).$$

Using formulas (38), it is possible to prove by induction that the function Φ_n can be written in the form $\Phi_n(s,t)=(s+t)P_n(st)+Q_n(st)$, where P_n and Q_n are polynomials. These polynomials obey the linear recurrence:

$$\begin{pmatrix} P_{n+1}(x) \\ Q_{n+1}(x) \end{pmatrix} = \tag{40}$$

$$\begin{pmatrix} 2a(1-a)x & a(2-a)(1-a)^2 \\ 2x((1-a)^2+a^2x) & (1-a)^4+a^2(2-a)^2x \end{pmatrix} \begin{pmatrix} P_n(x) \\ Q_n(x) \end{pmatrix},$$

with initial conditions $P_0(x)=0$ and $Q_0(x)=1$. The solution of this recurrence using the technique of section 4.1 leads, thanks to MAPLE, to:

$$EX_1(n)=\gamma n+\beta\left(1-(2a(a-1)(a^2-3a+1))^n\right)\;,$$

with γ given in (39), and

$$\beta=\frac{a(a-1)^2(a-2)(2a^2-2a+1)}{(1+2a(a-1)(a^2-3a+1))^2}\;.$$

Formula (39) follows because the term $\epsilon(a)=2a(a-1)(a^2-3a+1)$ is always less than 1 in modulus. Note that $EX_1(n)$ is actually a polynomial in a although it may not seem at first glance. Observe also that $\epsilon(a)$ changes sign at $a_0=(3-\sqrt{5})/2$. The generating function of the RV $\Delta(n)$ is given by $\Phi_n(s,\frac{1}{s})=P_n(1)(s+\frac{1}{s})+Q_n(1)$. Given the expressions for P_n and Q_n, it turns out that:

$$\begin{aligned} P_n(1) &= \frac{a(2-a)(1-a)^2}{1+2a(a-1)(a^2-3a+1)}\left(1-(2a(a-1)(a^2-3a+1))^n\right) \\ Q_n(1) &= \frac{(2a^2-2a+1)+2a(2-a)(1-a)^2(2a(a-1)(a^2-3a+1))^n}{1+2a(a-1)(a^2-3a+1)}, \end{aligned}$$

from which the result follows. Note that $\Delta(n)$ does not depend on $n\geq 1$ when $a=a_0$, which means that the system couples with its stationary regime at time $n=1$. Finally, if $a=1/2$, one has:

$$\mathbb{P}(\Delta(n)=\pm 1)=\frac{3}{14}(1-8^{-n}),\qquad \mathbb{P}(\Delta(n)=0)=\frac{4}{7}+\frac{3}{7}8^{-n}\;.$$

□

5.2 The general Bernoulli case

Let a, b, c and d denote the Bernoulli parameters of the four input sequences. It will actually be more convenient to use $\bar{a} = 1-a, \bar{b} = 1-b, \bar{c} = 1-c$ and $\bar{d} = 1-d$. We now have:

$$\sum_{x,y\geq 0} S(x+u, y+u, u, v) s^x t^y$$
$$= \left((1-a)(1-c+c\mathbf{1}_{\{u>v\}}) + \tfrac{s}{1-s}\right)\left((1-b)(1-d+d\mathbf{1}_{\{u>v\}}) + \tfrac{t}{1-t}\right).$$

The other sums involved in $B_n(s,t)$ and $C_n(s,t)$ have a similar form.

The generating function Φ_n satisfies the recurrence:

$$\begin{array}{rcll} \Phi_{n+1}(s,t) & = & (\bar{a}+as)(\bar{b}+bt) & A_n(st,1) \\ & + & (\bar{a}\bar{c}s + (1-\bar{a}\bar{c}))(\bar{b}\bar{d}t + (1-\bar{b}\bar{d})) & B_n(st) \\ & + & (\bar{c}+cs)(\bar{d}+dt) & C_n(1,st)\ , \end{array} \tag{41}$$

with initial condition $\Phi_0(s,t) = 1$. Computing the first values of Φ_n, one quickly realizes that this function must have the form:

$$\Phi_n(s,t) \;=\; P_n(st) \;+\; sQ_n(st) \;+\; tR_n(st)\ ,$$

where as usual, P_n, Q_n and R_n are polynomials. Moreover, the computations of A_n, B_n and C_n give simply:

$$A_n(x,1) = Q_n(x), \qquad C_n(x,1) = R_n(x), \qquad B_n(x) = P_n(x)\ .$$

Therefore, (41) translates into the linear recurrence for the vector $Y_n(x) = (P_n(x), Q_n(x), R_n(x))$: $Y_{n+1}(x) = M(x)Y_n(x)$ with

$$\begin{pmatrix} \bar{a}\bar{b}\bar{c}\bar{d} + x(1-\bar{a}\bar{b}(1-\bar{b}\bar{d}) & x(abx+\bar{a}\bar{b}) & x(cdx+\bar{c}\bar{d}) \\ \bar{b}\bar{d}(1-\bar{a}\bar{c}) & a\bar{b}x & c\bar{d}x \\ \bar{a}\bar{c}(1-\bar{b}\bar{d}) & b\bar{a}x & d\bar{c}x \end{pmatrix}\ ,$$

with $Y_0(x) = Y_0 = (1,0,0)^t$. Is it still necessary to tell how to solve this recurrence? Let us rather derive the Lyapunov exponent, following the reasoning of section 4.2. With $\mathcal{Y}(z,x) = \sum_{n=0}^{\infty} Y_n(x) z^n$, we have:

$$\mathcal{Y}(z,x) = \frac{[I - zM(x)\tilde{]}Y_0}{(1-z\lambda_1(x))(1-z\lambda_2(x))(1-z\lambda_3(x))}\ .$$

Let $u = (1,1,1)$. From the fact that $\Phi_n(1,1) = P_n(1) + Q_n(1) + R_n(1) = 1$, and because the matrix $M(1)^t$ is stochastic, we have:

$$u\mathcal{Y}(z,0) = \frac{1}{1-z} = \frac{u[I - zM(1)\tilde{]}Y_0}{(1-z)(1-z\lambda_2(1))(1-z\lambda_3(1))}\ ,$$

so that $u[I - zM(1)\tilde{]}Y_0 = (1-z\lambda_2(1))(1-z\lambda_3(1))$. Now, $EX_1(n) = d\Phi(s,1)/ds|_{s=1}$ $= P_n'(1) + Q_n'(1) + R_n'(1) + Q_n(1)$. Clearly, $Q_n(1) = o(n)$, so we just have to look at:

$$
\begin{aligned}
&(P_n + Q_n + R_n)'(1) \\
&= [z^n] u \frac{\partial \mathcal{Y}}{\partial x}(z,1) = [z^n] \frac{z\lambda_1'(1)\ u[I - zM(1)\tilde{]}Y_0}{(1-z)^2(1-z\lambda_2(1))(1-z\lambda_3(1))} + O\left(1/(1-z)\right) \\
&= n\lambda_1'(1)(1+o(n)) \ .
\end{aligned}
$$

We can again use theorem 2.2 or theorem 2.3 to compute $\lambda_1'(1)$. A MAPLE program based on these formulas eventually gives:

PROPOSITION 5.2 *The Lyapunov exponent of the system is given by:*

$$
\gamma = 1 - \frac{\delta}{\bar{a}} \bar{d}(\bar{b}\bar{c}(1-\bar{a})(1-\bar{d}) - (1-(1-\bar{b})(1-\bar{c}))^2) \ ,
$$

with

$$
\begin{aligned}
\delta = \ & 1 - 2\bar{a}\bar{b}\bar{c} + 2\bar{a}\bar{c}\bar{d} - 2\bar{a}\bar{b}\bar{c}\bar{d} - \bar{a}\bar{d} - 2\bar{b}\bar{c}\bar{d} + \bar{a}\bar{c} + \bar{b}\bar{d} + \bar{a}\bar{b} + \bar{c}\bar{d} - \bar{c} - \bar{b} + \\
& \bar{c}\bar{b} + \bar{a}^2\bar{c}\bar{b} - \bar{a}\bar{c}^2\bar{d} + 2\bar{a}\bar{d}\bar{b} + \bar{d}\bar{b}^2\bar{a}\bar{c} - \bar{d}^2\bar{b}\bar{a}\bar{c} + \bar{a}\bar{c}^2\bar{b}\bar{d} - \bar{a}^2 \\
& + \bar{c}\bar{b}\bar{d} - \bar{b}^2\bar{d}\bar{a} + \bar{b}\bar{d}^2\bar{c} \ .
\end{aligned}
$$

Finally, the generating function of the RV $\Delta(n)$ is: $\Delta_n^*(u) = P_n(1) + uQ_n(1) + u^{-1}Q_n(1)$, which is to say that this distribution is (as expected) concentrated on $\{-1, 0, 1\}$. The values of the masses are extracted from the generating vector $\mathcal{Y}(z,1)$ and are:

$$
Y_n(1) = [z^n] \frac{[I - zM(1)\tilde{]}Y_0}{(1-z)(1-z\lambda_2(1))(1-z\lambda_3(1))} \ .
$$

The actual computations are left to the reader. When $n \to \infty$, this vector converges exponentially fast to

$$
Y_\infty(1) = \frac{[I - M(1)\tilde{]}Y_0}{(1-\lambda_2(1))(1-\lambda_3(1))} \ .
$$

The rate of convergence is given by $|\lambda_2(1)|$, the modulus of the second eigenvalue of $M(1)$.

5.3 The symmetric geometric case

In this section, it is assumed that the four input sequences have geometric distributions with parameter a. Therefore, $P(\sigma = n) = (1-a)a^n$.

$$
\sum_{x,y \geq 0} S(x+u, y+u, u, v) s^x t^y \ =
$$

$$
\frac{(1-a)^2}{(1-as)(1-at)} \left(\frac{1}{1-s} - \frac{a^{u-v+1}}{1-a^2 s} \right) \left(\frac{1}{1-t} - \frac{a^{u-v+1}}{1-a^2 t} \right) \ .
$$

Using the symmetry of the problem, this yields:

$$A_n(s,t) = C_n(s,t) = \frac{(1-a)^2}{(1-as)(1-at)(1-a^2s)(1-a^2t)}$$
$$\big((1-a^2s)(1-a^2t)J_n(st,1) - ((1-s)(1-a^2t) + (1-t)(1-a^2t))J_n(ast,\tfrac{1}{a})$$
$$+(1-s)(1-t)J_n(a^2st,\tfrac{1}{a^2}).$$

The analysis is extremely similar to that of the symmetric exponential case. One shows by induction that

$$\Phi_n(s,t) =$$

$$\frac{1}{(1-as)(1-at)(1-a^2s)(1-a^2t)} \frac{(1-a)^{4n}}{D(st)^{n-1}} \left(P_n(st) + (s+t)Q_n(st)\right) ,$$

with $D(x) = (1-ax)(1-a^2x)^2(1-a^3x)^2(1-a^4x)^2$, and the polynomials P_n, Q_n given by a linear recurrence involving a 2×2 matrix too complicated to be written here. Symbolic manipulations eventually lead to the following result:

PROPOSITION 5.3 *The Lyapunov exponent of the system is given by the formula:*

$$\gamma = \frac{n(a)}{d(a)}$$

with

$$n(a) = a(a^{13} + 11a^{12} + 32a^{11} + 77a^{10} + 137a^9 + 218a^8 + 273a^7 + 289a^6$$
$$+244a^5 + 175a^4 + 99a^3 + 50a^2 + 18a + 4)$$
$$d(a) = (1-a)(a+1)(a^2+a+1)$$
$$(a^{10} + 6a^9 + 8a^8 + 20a^7 + 25a^6 + 32a^5 + 25a^4 + 20a^3 + 8a^2 + 6a + 1).$$

It is possible to check that when the parameters a and the time unit δ are chosen in such a way that the geometric distribution approaches the exponential distribution, namely:

$$\delta\frac{a}{1-a} = 1,$$

with $\delta \to 0$, then the Lyapunov exponent of the system tends to 407/228.

ACKNOWLEDGEMENT

We are extremely thankful to O. Boxma for his very careful reading of the earlier version of this manuscript.

A JOINT DISTRIBUTIONS AND TRANSFORMS

We state in this appendix a number of formulas useful for computing distributions of maxima (or minima) of random variables, when their joint distribution is known in transformed form: Laplace transform or z-transform (i.e. the generating function).

We provide formulas both for the continuous and the discrete case.

This section is essentially an excerpt of [8], adapted to the computations of the paper. Such formulas also appear in the works of J.W. Cohen: see for instance [4, p. 564] for the discrete case. The technique relies on the following lemma of the complex variable calculus.

LEMMA A.1 *Let L be a smooth contour of $\mathbb{C}$, which separates $\mathbb{C}$ in two domains L^+ (to the left) and L^- (to the right). Let ϕ be a function defined on L satisfying Hölder's condition. Assume a function Φ, defined on $\mathbb{C}\backslash L$ is analytic on L^+ and L^- and admits a limit when $z \to t \in L$ both in L^+ and L^- and is such that:*

$$\forall t \in L, \quad \lim_{z \to t, z \in L^+} \Phi(z) - \lim_{z \to t, z \in L^-} \Phi(z) = \phi(t) .$$

If furthermore $\Phi(z)$ vanishes when $|z| \to \infty$, then

$$\forall z \in \mathbb{C}, \qquad \Phi(z) = \frac{1}{2i\pi} \int_L \frac{\phi(t)}{t - z} dt .$$

The proof of this result may be found in [5] in the case of a closed contour. The proof for a smooth open contour can be found in [8]. Note that the contour has to be smooth at infinity, that is be sufficiently regular and tend to infinity in opposite directions.

THE CONTINUOUS CASE Let A and B be two real RVs, not necessarily positive nor independent. Assume one knows their joint distribution by its Laplace transform:

$$G(x, y) = \mathbb{E}(e^{-xA - yB}) \qquad \forall\, \Re(x) = 0,\ \Re(y) = 0 .$$

Note that this function is defined only on $i\mathbb{R} \times i\mathbb{R}$ since A and B may be negative.

Let

$$\begin{aligned} J(s,t) &= \mathbb{E}(e^{-sA - tB} \mathbf{1}_{\{A > B\}}) \\ K(s) &= \mathbb{E}(e^{-sA} \mathbf{1}_{\{A = B\}}) \\ L(s,t) &= \mathbb{E}(e^{-sA - tB} \mathbf{1}_{\{A < B\}}). \end{aligned}$$

These functions are defined on the domains $\{(s,t) | \Re(s+t) = 0, \Re(s) \geq 0\}$, $\{\Re(s) = 0\}$ and $\{(s,t) | \Re(s+t) = 0, \Re(t) \geq 0\}$, respectively. They can be computed from G, as stated in the following lemma.

LEMMA A.2 *If for any $x \in i\mathbb{R}$ the function $y \mapsto G(x+y, -y)$ admits a limit when $|y| \to \infty$, then:*

$$J(s,t) = -\frac{1}{2i\pi}\int_{i\mathbb{R}} G(s+t+z,-z)\frac{dz}{z+t} - \frac{1}{2}K(s+t) \qquad \Re(t)<0, \Re(s+t)=0$$

$$K(x) = -2\ \lim_{|t|\to\infty,\Re(t)<0} \frac{1}{2i\pi}\int_{i\mathbb{R}} G(x+z,-z)\frac{dz}{z+t} \qquad \Re(s)=0$$

$$L(s,t) = \frac{1}{2i\pi}\int_{i\mathbb{R}} G(s+t+z,-z)\frac{dz}{z+t} - \frac{1}{2}K(s+t) \qquad \Re(t)>0, \Re(s+t)=0\ .$$

PROOF Fix $x \in i\mathbb{R}$. Define the function Φ_x on the domain $\mathbb{C}\setminus i\mathbb{R}$ by:

$$\Phi_x(y) = \begin{cases} -\ \mathbb{E}(e^{-xA}e^{-y(A-B)}\ \mathbf{1}_{\{A-B>0\}}) & \text{if } \Re(y)>0 \\ \mathbb{E}(e^{-xA}e^{-y(A-B)}\ \mathbf{1}_{\{A-B<0\}}) & \text{if } \Re(y)<0\ . \end{cases}$$

One easily checks that

$$\lim_{z\to y,\Re(z)<0} \Phi_x(z) - \lim_{z\to y,\Re(z)>0} \Phi_x(z) = G(x+y,-y) - K(x) \quad \forall\, y \in i\mathbb{R}.$$

The function $\Phi_x(y)$ vanishes at infinity in all directions other than $i\mathbb{R}$. If moreover the mapping $y \mapsto G(x+y,-y)$ satisfies Hölder's condition on $i\mathbb{R}$ and admits a limit when $|y| \to \infty$, we can apply lemma A.1 to obtain:

$$\begin{aligned}\Phi_x(z) &= \frac{1}{2i\pi}\int_{i\mathbb{R}} (G(x+y,-y)-K(x))\frac{dy}{y-z} \\ &= \frac{1}{2i\pi}\int_{i\mathbb{R}} G(x+y,-y)\frac{dy}{y-z} - sgn(z)\ K(x), \end{aligned} \tag{42}$$

for any $z \in \mathbb{C} \setminus i\mathbb{R}$, where the sign function is defined below (see lemma A.5). This gives the value of $L(s,t) = \Phi_{s+t}(-t)$ and $J(s,t) = -\Phi_{s+t}(-t)$ for values of t in $\{\Re(t)>0\}$ and $\{\Re(t)<0\}$ respectively. The value of $K(x)$ is obtained from (42) by taking the limit: by definition, $\Phi_x(-t)$ vanishes as $|t| \to \infty$.

Note finally that a change of variables gives an alternate formula for $L(s,t)$, which makes the symmetry between A and B appear:

$$L(s,t) = -\frac{1}{2i\pi}\int_{i\mathbb{R}} G(-z,s+t+z)\frac{dz}{z+s} - \frac{1}{2}K(s+t),$$

for $\Re(s)<0$ and $\Re(s+t)=0$.

This lemma has the following corollary:

LEMMA A.3 *Let* $S^*(s) = \mathbb{E}(s^{\max(A,B)})$. *It is given by:*

$$S^*(s) = - \lim_{\substack{z \to 0 \\ \Re(z) > 0}} \frac{1}{2i\pi} \left[\int_{i\mathbb{R}} G(s+y,-y) \frac{dy}{y-z} + \int_{i\mathbb{R}} G(-y,s+y) \frac{dy}{y-z} \right] .$$

THE DISCRETE CASE Let X and Y be two discrete random variables, and let $G(x,y) = \mathbb{E}(x^X y^Y)$ be their joint generating function.

We are interested in computing the functions:

$$\begin{aligned} A(x,y) &= \mathbb{E}(x^X y^Y \mathbf{1}_{\{X>Y\}}) \\ B(x) &= \mathbb{E}(x^X \mathbf{1}_{\{X=Y\}}) \\ C(x,y) &= \mathbb{E}(x^X y^Y \mathbf{1}_{\{X<Y\}}) , \end{aligned}$$

which are defined on the domains $\{|x| \leq 1, |xy| = 1\}$, $\{|x| = 1\}$ and $\{|y| \leq 1, |xy| = 1\}$, respectively.

LEMMA A.4 *Let C be the unit circle of $\mathbb{C}$. We have:*

$$\begin{aligned} A(x,y) &= -\frac{1}{2i\pi} \int_C G(\frac{1}{z}, xyz) \frac{dz}{z-1/x} \qquad \text{for } |x| < 1, |xy| = 1 \\ B(x) &= \frac{1}{2i\pi} \int_C G(xz, \frac{1}{z}) \frac{dz}{z} \qquad \text{for } |x| = 1 \\ C(x,y) &= -\frac{1}{2i\pi} \int_C G(xyz, \frac{1}{z}) \frac{dz}{z-1/y} \qquad \text{for } |y| < 1, |xy| = 1. \end{aligned}$$

Consequently, the generating function of $\max(X,Y)$: $S^*(x) = \mathbb{E}(x^{\max(X,Y)})$, *is given, for any* $|x| = 1$, *by:*

$$S^*(x) = \frac{1}{2i\pi} \int_C G(xz, \frac{1}{z}) \frac{dz}{z} - \lim_{\substack{t \to x \\ |t|<1}} \frac{1}{2i\pi} \int_C \left(G(\frac{1}{z}, tz) + G(tz, \frac{1}{z}) \right) \frac{dz}{z-1/t} .$$

PROOF As in the proof of Lemma A.1, define, for any complex number x such that $|x| = 1$, the function Ψ_x by:

$$\Psi_x(y) = \begin{cases} \mathbb{E}(x^X y^Y \mathbf{1}_{\{X-Y \geq 0\}}) & \text{if } |y| < 1 \\ - \mathbb{E}(x^X y^Y \mathbf{1}_{\{X-Y<0\}}) & \text{if } |y| > 1 . \end{cases}$$

The function Ψ_x vanishes when $|y| \to \infty$, and satisfies:

$$\lim_{z \to y, |z|<1} \Psi_x(z) - \lim_{z \to y, |z|>1} \Psi_x(z) = G(xy, \frac{1}{y}) \quad \forall\, y \in C.$$

Lemma A.1 applies again, and gives:

$$\Psi_x(y) = \frac{1}{2i\pi}\int_{\mathcal{C}} G(xz, \frac{1}{z})\frac{dz}{z-y} .$$

This gives the value of $C(x,y) = -\Psi(xy, 1/y)$ for $|y| < 1$ and $|xy| = 1$. Exchanging X and Y in this computation yields the formula for $A(x,y)$. Finally, $B(x)$ is obtained as the limit of $\Psi_x(y)$ then $y \to 0$, whereas the formula for $S^*(x)$ follows from: $S^*(x) = A(x,1) + B(x) + C(1,x)$.

To conclude, we state the following lemma, which allows to make actual computations based on Lemmas A.2 and A.4. Define the "sign" function on $\mathbb{C} \setminus i\mathbb{R}$ by:

$$sgn(\zeta) = \frac{1}{2i\pi}\int_{i\mathbb{R}} \frac{dz}{z-\zeta} .$$

LEMMA A.5 *The value of the sign function is:*

$$sgn(\zeta) = \begin{cases} 1/2 & \text{if } \Re(\zeta) < 0 \\ -1/2 & \text{if } \Re(\zeta) > 0. \end{cases}$$

B A MAPLE PROGRAM FOR THE SEMI-SYMMETRIC EXPONENTIAL CASE

```
# MAPLE program to compute the Lyapunov exponent of the
# exponential, semi-symmetric system
#
Fract := 1/(a+s)/(b+s)/(a+b+s)/(b+t)/(a+t)/(b+a+t);

Intgd := subs( { s=x+y, t=-y }, Fract) / (y-z);

t1 := limit( Intgd*(x+y+a), y=-a-x );
t2 := limit( Intgd*(x+y+b), y=-b-x );
t3 := limit( Intgd*(x+y+a+b), y=-a-b-x );

J:=-subs({x=u+v,z=-v}, Na*t1+Nb*t2+Nab*t3);

J0:=subs({u=x,v=0},J);
Ja:=subs({u=x+a,v=-a},J);
Jb:=subs({u=x+b,v=-b},J);
Jab:=subs({u=x+b+a,v=-b-a},J);

M0 := (x*(a+b)+y+(a+b)^2)*(2*y+x*(a+b)+2*a*b);
Ma := -(x*b+y+b**2)*(2*y+x*(a+b));
Mb := -(x*a+y+a**2)*(2*y+x*(a+b));
Mab := y*(2*y+x*(a+b)+2*a*b);

NN := PP + y*QQ + y^2*RR;
Na := subs( y=-a*(x+a), NN );
Nb := subs( y=-b*(x+b), NN );
Nab := subs( y=-(a+b)*(x+(a+b)), NN );
```

```
NewP := (J0*M0 + Ja*Ma + Jb*Mb + Jab*Mab);

NumNewP := collect(numer(normal(NewP)),y);
PP1 := collect(coeff(NumNewP,y,0),{PP,QQ,RR});
QQ1 := collect(coeff(NumNewP,y,1),{PP,QQ,RR});
RR1 := collect(coeff(NumNewP,y,2),{PP,QQ,RR});

B_11 := coeff(PP1,PP,1); B_12 := coeff(PP1,QQ,1);
B_13 := coeff(PP1,RR,1); B_21 := coeff(QQ1,PP,1);
B_22 := coeff(QQ1,QQ,1); B_23 := coeff(QQ1,RR,1);
B_31 := coeff(RR1,PP,1); B_32 := coeff(RR1,QQ,1);
B_33 := coeff(RR1,RR,1);

B := matrix(3,3,[B_11,B_12,B_13,B_21,B_22,B_23,B_31,B_32,B_33]);

#
# A procedure to compute recursively P_n(x)
#
Pi := proc(n)
if n=1  then 2*x*(a+b)+(a+b)^2;
else Pi(n-1)*B_11+Qi(n-1)*B_12+Ri(n-1)*B_13;
fi;
end;

#
# The computation of the Lyapunov exponent, using the
#  characteristic polynomial of the matrix.
#
PolCar := collect(charpoly(B,z),z);

l10 := factor( subs( x=0 , B[1,1] ) );

corr := 2/a+2/b+2*(1/(a+b)+1/(2*a+b)+1/(a+2*b)+1/2/(a+b));

Chi := -simplify(subs({x=0,z=l10},diff(PolCar,x)) / diff(PolCar,z));

gamma := -simplify( Chi / l10 - corr );
```

References

1. F. Baccelli, "Ergodic Theory of Stochastic Petri Networks", *The Annals of Probability*, **20**(1992), 375–396.
2. F. Baccelli, A. Jean-Marie and Z. Liu, "A Survey on Solution Methods for Task Graph Models", Proceedings of the Second QMIPS Workshop, Erlangen, March 1993.

3. F. Baccelli and P. Konstantopoulos, "Large Deviation Estimates of Cycle Times in Stochastic Petri Nets", *Proc. Workshop Probabilities*, Princeton, 1992.
4. J.W. Cohen, *The Single Server Queue*, North-Holland, 1982.
5. J.W. Cohen and O. Boxma, *Boundary Value Problems in Queueing System Analysis*, North Holland 1983.
6. P. Flajolet and A. Odlyzko, "Singularity Analysis of Generating Functions", *SIAM J. Disc. Math.*, **3** No. 2 (1990), pp. 216–240.
7. L. Flatto and H.P. Mc Kean, "Two Queues in Parallel", *Comm. on Pure and Applied Math.*, **30** (1977), pp. 255–263.
8. A. Jean-Marie, *Aspects qualitatifs et quantitatifs des réseaux d'interconnexion multi-étages*, Thèse de 3ème cycle, Université Paris XI, Orsay, 1987.
9. A. Jean-Marie and G.-J. Olsder, "Analysis of Stochastic Min-Max Systems: Results and Conjectures", technical report #93-94, Delft University of Technology, The Netherlands, 1993.
10. J.A.C. Resing, R.E. de Vries, M.S. Keane, G. Hooghiemstra and G.J. Olsder, "Asymptotic Behavior of Random Discrete Event Systems", *Stochastic Processes and their Applications*, **36** (1990), pp. 195–216.
11. E. Seneta, *Non-negative Matrices and Markov Chains*, Springer Verlag, 2nd edition, 1981.
12. J.M. Vincent, "Some ergodic results on stochastic iterative DEDS systems", technical report of IMAG, University of Grenoble, France, 1994.

A Graphical Representation for Matrices in the (Max,+) Algebra

Jean Mairesse *
INRIA-Sophia Antipolis
B.P. 93, 06902 Sophia Antipolis Cedex, France

We study matrices in the $(Max,+)$ algebra. We introduce a new tool for describing the deterministic spectral behaviour of matrices of size 3×3. It consists in a graphical representation of eigenvectors and domains of attraction in the projective space. It appears to be very helpful in understanding some of the phenomena occurring in this algebra.

1 INTRODUCTION

Many communication or manufacturing networks can be represented by Discrete Events Dynamic Systems (DEDS). Recent researches have dealt with the problem of finding a unified framework to study DEDS. Petri Networks, and more precisely Event Graphs (EG), are an example of such a formalism. They model phenomena such as synchronization or blocking. These networks have an easy algebraic interpretation in a non conventional algebra. More precisely, it is possible to show that a timed Event Graph can be represented as a linear recursive equation in the (Max,+) algebra, of the following kind:

$$y_{n+1} = A \otimes y_n \ ,$$

where y_{n+1} and y_n are $\mathbb{R}^J$-valued vectors and A is a matrix of size $J \times J$. The matrix-vector product has to be interpreted in the (Max,+) algebra. For a timed Event Graph, the dimension J is the number of transitions. The vector y_n consists of the dates of the n^{th} firing of the transitions. For more insights on all modelling aspects, the reader is referred to [1] or [2].

The spectral theory of matrices in the (Max,+) algebra is now well known. It can be tracked back to [8] or, for the Russian school, to [10]. One of the main differences with the classical spectral theory is that there is a unique eigenvalue for irreducible matrices. As a consequence, the main interest and difficulty in the (Max,+) algebra is to study eigenvectors associated with the unique

*Research supported by the Direction des Recherches Etudes et Techniques (DRET) under contract n° 91 815. Supported in part by the European Grant BRA-QMIPS of CEC DG XIII.

eigenvalue. For a timed EG, the eigenvalue is exactly the mean cycle time (inverse of the throughput rate). On the other hand, eigenvectors are associated with quantities such as number of tokens in a place, waiting times or idle times. Multiple eigenvectors will mean multiple regimes for these quantities.

In this paper, we present the classical spectral results under a new light. We develop a tool for describing the spectral behaviour of matrices of size 3×3. It consists in a graphical representation of asymptotic regimes in the projective space. This representation enables us to get an intuition of the spectral behaviour of larger matrices. It appears also very useful in order to understand some phenomena occurring in this algebra, especially in stochastic systems.

The paper is organized as follows. In Section **2**, we propose some motivating examples and models. Sections **3** and **4** review some basic results on the (Max,+) algebra and its spectral theory respectively. In Section **4.3**, we present also a complete spectral analysis of matrices of size 3 with the help of the graphical representation mentioned before. Section **5** is devoted to two examples of utilizations of this graphical representation. In the first example, we give a "visual" example of a projectively infinite semigroup of matrices. The second one shows how the graphical representation can be used for stochastic models.

2 Some Motivating Models

We consider systems whose dynamic behaviour is driven by a recursive equation of the form:

$$y_i(n+1) \quad = \quad \max_{1 \leqslant j \leqslant J} (A_{ij} + y_j(n)),\ i = 1, \ldots, J\,. \tag{1}$$

We allow A_{ij} to be equal to $-\infty$. We say that the matrix $A = (A_{ij}, i, j = 1, \ldots, J)$ is irreducible if $\forall i, j\ \exists (i_1 = i, i_2, \ldots, i_n = j)\ s.t.\ A_{ii_2} + A_{i_2 i_3} + \cdots + A_{i_{n-1} j} > -\infty$.

We define the notations:

$$\mathbb{R}_\star = \mathbb{R} \cup \{-\infty\}\,.$$

$$\forall x, y \in \mathbb{R}_\star,\ \ x \oplus y = max(x, y),\ x \otimes y = x + y\,.$$

With these notations, the basic evolution equation (1) takes a very simple and convenient form. In fact, it can be rewritten as:

$$y(n+1) \quad = \quad A(n) \otimes y(n)\,. \tag{2}$$

Here $y(n) = (y_1(n), y_2(n), \ldots, y_J(n))'$ and the matrix-vector product is defined in the natural way just by replacing $+$ and $\times$ by $\oplus$ and $\otimes$ (i.e. $(A \otimes y)_i = Max_k(A_{ik} + y_k) = \bigoplus_k A_{ik} \otimes y_k$).

In this section we use this matrix notation only for the ease of presentation. But in fact, as we will see in a moment, it is more than a simple notation game. Many results from classical linear algebra can be transferred to equations like (2) which are linear with respect to operations max and +.

We are going to propose several illustrating examples of such systems. The graphical representation introduced in this paper corresponds to deterministic and irreducible systems of dimension 3 (J=3). As a consequence the different examples we are going to propose will also be deterministic, non-autonomous (irreducible) and of dimension 3. For most of the systems to be presented, there exist analogous systems corresponding to non-irreducible cases. The techniques of this paper can in this case be applied to "irreducible sub-systems". Of course larger systems can be considered as well. At last, and in all cases, there exist natural stochastic extensions.

For systems verifying Equation (1), we can consider two kinds of asymptotic results.

- First order limits, on ratios:

$$\lim_k \frac{||y(k)||}{k}\ ,\quad \lim_k \frac{y_i(k)}{k}\ .$$

- Second order limits, on differences:

$$\lim_k\ y_i(k+1) - y_i(k),\ \forall i, \lim_k\ y_j(k) - y_i(k),\ \forall i \neq j\ .$$

These quantities are closely related to the solutions of an eigenvalue problem defined in the following way. We want to find non trivial solutions of the equation:

$$\max_{1 \leqslant j \leqslant J}(A_{ij} + y_j) \;=\; \lambda + y_i,\ i = 1, \ldots, J \tag{3}$$

$$\text{or } A \otimes y \;=\; \lambda \otimes y\ , \tag{4}$$

where $A \in \mathbb{R}^{J \times J}$ is a matrix, y is a column vector (the "eigenvector") and λ is a real constant (the "eigenvalue"). For a given matrix A, first order limits as defined above are eigenvalues of A. Second order limits can be expressed in terms of the eigenvectors of A. In this paper, we focus essentially on second order results, i.e. on eigenvectors.

From a less algebraic point of view, we will see in several examples how to interpret first and second order limits respectively.

2.1 Parallel programs

We consider a parallel program executed on several identical processors. We model it by means of its precedence graph or developed graph. If we consider a system of J processors, the graph τ has a set of nodes which is $J \times \mathbb{N}$. Node (i, n) represents the n^{th} task at processor i. The arcs between nodes represent the precedence constraints. There is an arc between node (i, n) and

node (j,m) (denoted $(i,n) \rightarrow (j,m)$) if the n^{th} task at processor i has to be performed in order for the m^{th} task at processor j to be enabled. A task is enabled if and only if all the activities of its incoming arcs are completed. Activity begins as soon as the task is enabled. Each activity has a duration which depends on the processor. In Figure 1, example (A), we have represented the task graph of three processors having no communications. Each processor is working sequentially and has to complete task n before beginning task $n+1$. In example (B) we have represented a parallel program with synchronizations between processors, implying additional arcs for the task graph.

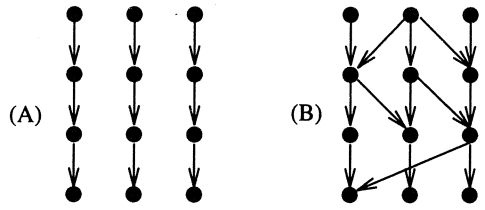


FIGURE 1. Precedence graphs of parallel programs running on three processors

It is clear that a precedence graph is live (i.e. without deadlock) if and only if it is acyclic. Otherwise there would be a cycle of precedence constraints implying that a task has to be completed in order to be enabled!

A precedence graph is irreducible if

$$\forall i\ \forall j\ \exists n\ \exists (i_1 = i, \ldots, i_n = j)\ \exists (k_1, \ldots, k_n) \text{ s.t.}$$

$$(i,k_1) \rightarrow (i_2,k_2) \rightarrow \cdots \rightarrow (j,k_n)\,.$$

It is periodic of period d if

$$(i,n) \rightarrow (j,m) \Longrightarrow (i,n+d) \rightarrow (j,m+d)\,.$$

From now on, we consider parallel programs whose precedence graphs are acyclic, irreducible and periodic.
We assume first that the period is 1 and that synchronization arcs exist only between level n (i.e. nodes $(1,n),\ldots,(J,n)$) and level n+1. We denote by $y_i(n)$ the date of completion of task n at processor i, and by A_{ij} the duration of the synchronization constraint between nodes (i,n) and $(j,n+1)$ (it may include a transmission time as well as the activity time at processor j). If there is no synchronization between (i,n) and $(j,n+1)$, we set $A_{ij} = -\infty$. It is now obvious that Equation (1) describes the dynamics of the system.

If the period is 1 without any further restrictions, then it is possible to come back to the previous case through a renumbering of nodes and an expansion of the dimension of the system in Equation (1). An interesting problem is then to find a minimal representation of the system which means precisely a (Max,+)

linear system of the form of Equation (1) of minimal dimension.

If the period of the precedence graph is $d > 1$ then the dynamics of the system can still be written as a (Max,+) recursive equation of the form of (1). But the dimension of the problem is now at least $J \times d$, one level containing at least all the nodes of a period of the initial graph.

First order quantities, i.e. $\lim y_i(n)/n$ correspond to the average execution time of a task on a processor. From second order quantities, one can compute delays between processors, idle times of processors or waiting times of tasks. Let us consider one example:

$$z = y_i(n+1) - (y_i(n) + A_{ii}) .$$

The real z is the idle time of processor i, i.e. the time during which processor i is ready to operate but is waiting for other processors. To have a better idea of the vast literature existing on the subject, see [3] and the references there.

2.2 Manufacturing Model

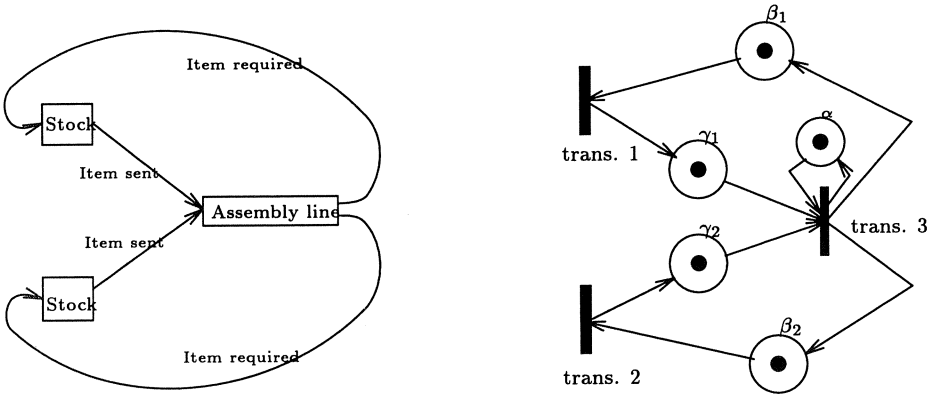


FIGURE 2. A manufacturing model and its Petri Net representation.

There are two types of items which have to be assembled together to form a part. There is a stock for each kind of item. As we are interested in autonomous systems, we assume that these stocks are infinite. Each time a part is completed at the assembly line, a request is sent to the storage facilities. New items are then sent to the assembly line. We denote:

- α: operating time at the assembly line.
- $\beta_i, i = 1, 2$: communication time between the assembly line and stock i.
- $\gamma_i, i = 1, 2$: transportation time between stock i and the assembly line.

We have represented in Figure 2 the Event Graph corresponding to this model. We consider three daters (y_i, $i = 1, 2, 3$) associated with this system. The first two correspond to the dates at which an item is sent from the stocks. The third one corresponds to the dates at which a part is completed at the assembly line. The (Max,+) linear system corresponding to this Event Graph is the following one:

$$y(n+1) = A \otimes y(n),\ A = \begin{pmatrix} 0 & \varepsilon & \beta_1 \\ \varepsilon & 0 & \beta_2 \\ \gamma_1 + \alpha & \gamma_2 + \alpha & \alpha \end{pmatrix},$$

with the notation $\varepsilon = -\infty$. For this model, first order quantities correspond to the mean cycle time, the inverse of the throughput. From second order quantities, one can compute for example the idle time of the assembly line between the completion of a task and the beginning of the next one. Let us denote it by δ.

$$\delta(n) = y_3(n) - y_3(n-1) - \alpha\,.$$

Of course much more complicated manufacturing systems can be modelled using Event Graphs of larger dimension. We can mention job-shop models or models using Kanban regulation. A vast literature exists on this topic, see [2] or [4].

The manufacturing system we have considered can be modelled using an Event Graph representation as shown in Figure 2. Event Graphs also called Marked Graphs or Decision Free Petri Nets are a special class of Petri Nets. They can efficiently model systems with synchronization, fork-join properties and/or blocking. On the other hand they can not deal with decision and routing. It has been proved in [1] that all Event Graphs can be described by an evolution equation of the form of equation (1).

3 The (Max,+) Algebra

DEFINITION 3.1 ((MAX,+) ALGEBRA) *We consider the semi-field (improperly called algebra)* $(\mathbb{R}^\star, \oplus, \otimes)$, *where* $\mathbb{R}^\star = \mathbb{R} \cup \{-\infty\}$. *The law* $\oplus$ *is "Max" and* $\otimes$ *is the usual addition. We set* $\varepsilon = -\infty$ *and* $e = 0$. *The element* ε *is neutral for the operation* $\oplus$ *and absorbing for* $\otimes$. *The element* e *is neutral for* $\otimes$. *The law* $\oplus$ *is idempotent, i.e.* $a \oplus a = a$. $(\mathbb{R}^\star, \oplus, \otimes)$ *is an idempotent semiring, called a dioid. It is moreover a commutative dioid. We shall write it* $\mathbb{R}_{Max}$.

In the rest of the paper, the notations "+,×" will stand for the usual addition and multiplication. Nevertheless, we will write ab for $a \otimes b$ whenever there is no possible confusion.

We define the product spaces $\mathbb{R}^J_{Max}$, $\mathbb{R}^{J\times J}_{Max}$. We define the product of a vector by a scalar: $a \in \mathbb{R}_{Max}, u \in \mathbb{R}^J_{Max}$, $(a \otimes u)_i = a \otimes u_i$.
Matrix product is defined in the following way. Let $A, B \in \mathbb{R}^{J\times J}_{Max}$,

$$(A \otimes B)_{ij} = Max_k(A_{ik} + B_{kj}) = \bigoplus_k A_{ik} \otimes B_{kj} \, .$$

Matrix-vector or scalar-matrix products are defined in a similar way.

We are interested in an eigenvalue problem in $\mathbb{R}_{Max}$, similar to the one of the traditional linear algebra. We want to find non trivial solutions to the equation:

$$A \otimes x = \lambda \otimes x \, ,$$

where $A \in \mathbb{R}^{J \times J}$ is an irreducible (see definition 4.2) matrix, x is a column vector (the "eigenvector") and λ is a real constant (the "eigenvalue"). We also define periodic solutions of the eigenvalue problem.

DEFINITION 3.2 *A periodic solution of period d is a set of vectors $\{x_1, \ldots, x_d\}$ of $\mathbb{R}^J$ verifying $Ax_i = \lambda x_{i+1}$, $i = 1, \ldots, d-1$ and $Ax_d = \lambda^d x_1$.*

Remark A periodic solution of period d for A implies the existence of d eigenvectors for A^d.

First of all, let us introduce the graphical representation that we are going to use extensively.

DEFINITION 3.3 ($\mathbb{PR}^J_{Max}$) *The projective space $\mathbb{PR}^J_{Max}$ is defined as the quotient of $\mathbb{R}^J_{Max}$ by the parallelism relation:*

$$u, v \in \mathbb{R}^J \quad u \simeq v \Longleftrightarrow \exists a \in \mathbb{R}_{Max} \setminus \{\varepsilon\} \text{ such that } u = a \otimes v \, .$$

Let π be the canonical projection of $\mathbb{R}^J_{Max}$ into $\mathbb{PR}^J_{Max}$.

In the rest of the paper, we will concentrate on aperiodic matrices (see definition 4.2). In this case, a matrix A, which is a linear operator of $(\mathbb{R}^J_{Max}, \oplus, \otimes)$, can be restricted to an operator of $(\mathbb{R}^J, \oplus, \otimes)$ (i.e. if u is a vector whose coordinates are all different from ε, then Au has the same property). As a consequence, we will consider only vectors in $\mathbb{R}^J$ and their projection in $\mathbb{PR}^J$, where $\mathbb{PR}^J$ is defined exactly in the same way as $\mathbb{PR}^J_{Max}$.

The canonical projection π of $\mathbb{R}^J$ into $\mathbb{PR}^J$ can be interpreted geometrically. It is nothing else than the orthogonal projection on the hyperspace orthogonal to the vector $\mathbb{1} = (1, \ldots, 1)'$. The projective space $\mathbb{PR}^J$ is isomorphic to $\mathbb{R}^{J-1}$. Let us consider a deterministic matrix $A \in \mathbb{R}^{J \times J}_{Max}$ and the $\mathbb{R}_{Max}$ eigenvalue problem $Ax = \lambda x$. For matrices of size 2 or 3, a graphical representation is possible in $\mathbb{R} \simeq \mathbb{PR}^2$ and $\mathbb{R}^2 \simeq \mathbb{PR}^3$ respectively. We represent eigenvectors and periodic regimes modulo the parallelism relation. Let us illustrate this. Figure 3 corresponds to the matrix

$$A = \begin{pmatrix} 2 & e \\ 1 & 2 \end{pmatrix} .$$

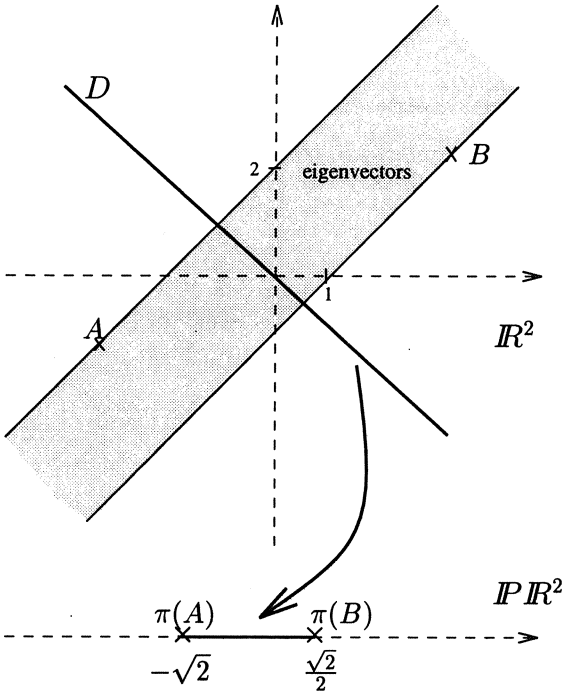


FIGURE 3. Dimension 2. A scs2-cyc1 matrix.

As we will see in a moment, the spectral theory tells us that there is a strip (an interval in $\mathbb{P}\mathbb{R}^2$) of eigenvectors and no periodic regimes of period greater than 1. The line D is the hyperspace orthogonal to the first bisecting line. We consider the orthogonal projection of the picture on D.

From now on, we will consider mostly matrices of size 3 whose spectral behaviour is much richer and can be graphically represented in $\mathbb{R}^2 \simeq \mathbb{P}\mathbb{R}^3$.

Let us introduce a distance on $\mathbb{P}\mathbb{R}^J_{Max}$ which we are going to call the projective distance.

DEFINITION 3.4 *We consider $x, y \in \mathbb{P}\mathbb{R}^J$. Let $u, v \in \mathbb{R}^J$ be two representatives of x and y respectively, i.e. $\pi(u) = x$ and $\pi(v) = y$.*

$$d(x,y) = d(u,v) = \bigoplus_i (u_i - v_i) \oplus \bigoplus_i (v_i - u_i)\,.$$

It is easy to check that $d(x,y)$ does not depend on the representatives u and v. It is also easy to check that it is a distance in $\mathbb{P}\mathbb{R}^J$. It is nothing else than the $\mathbf{L}_\infty$ norm on the projective space $\mathbb{P}\mathbb{R}^J$. We write either $d(x,y)$ or $d(u,v)$ with a little abuse of notation. We have the following property.

PROPOSITION 3.1 *Let A be an irreducible matrix of size J. Let u, v be two vectors of $\mathbb{R}_{Max}^J$. We have:*

$$d(Au, Av) \leq d(u, v) .$$

There is no simple criterion to get a strict inequality.

Let us represent the unit ball of the projective distance in $\mathbb{P}\mathbb{R}^3$.

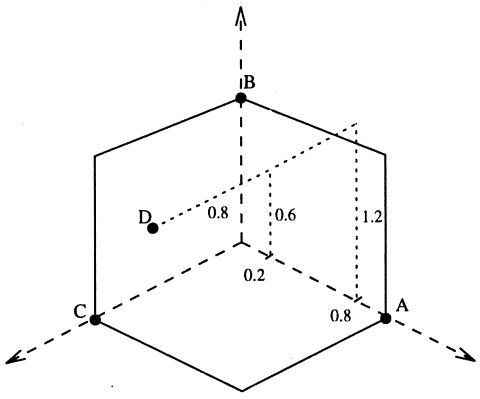


FIGURE 4. Unit ball of the projective distance

The regular hexagon in Figure 4 is the intersection of the unit square (i.e. the unit ball of $\mathbb{R}^3$ for the $\mathbf{L}_\infty$ norm) and the projection plane. The three represented axes are the orthogonal projection of the basis of $\mathbb{R}^3$. The represented points are:

$$A = \pi \begin{pmatrix} 1 \\ e \\ e \end{pmatrix}, \; B = \pi \begin{pmatrix} e \\ 1 \\ e \end{pmatrix}, \; C = \pi \begin{pmatrix} e \\ e \\ 1 \end{pmatrix}, \; D = \pi \begin{pmatrix} 0.2 \\ 0.6 \\ 0.8 \end{pmatrix} .$$

The practical way of representing a point X of $\mathbb{P}\mathbb{R}^3$ is to choose a vector ($\in \mathbb{R}^3$) in the parallelism class of X and to draw it in the three axes obtained by projection of the orthonormal basis of $\mathbb{R}^3$ (it is easy to check that the point we obtain does not depend on the representative in the parallelism class). The point D of Figure 4 illustrates this, we have drawn two constructions: one corresponding to (0.2, 0.6, 0.8) and the other one to (0.8, 1.2, 1.4) = $0.6 \otimes (0.2, 0.6, 0.8)$.

4 AN ILLUSTRATED SPECTRAL THEORY

We are now ready to review the $\mathbb{R}_{Max}$ spectral theory of irreducible matrices. The results we are going to present are now classical. A complete treatment can be found in [2]. For the spectral theory of reducible matrices, the reference is [6]. The analog theory for non finite dimensions is exposed in [5]. However, the idea of illustrating the spectral behaviour by graphical representations in the projective space is new.

4.1 General Presentation

From now on, we consider only irreducible matrices in $\mathbb{R}_{Max}^{J\times J}$. We recall that we want to find non trivial solutions to the equation $Ax = \lambda x$. Let us recall some definitions adapted to the $\mathbb{R}_{Max}$ algebra.

DEFINITION 4.1 *The communication graph of a square matrix A is a directed graph having a number of nodes equal to the size of A. This graph contains an arc from i to j iff $A_{ji} \neq \varepsilon$. The valuation of this arc is A_{ji}.*

DEFINITION 4.2 *A square matrix A is irreducible if: $\forall i, j \ \exists m \geq 0 \mid (A^m)_{ij} > \varepsilon$ (or equivalently if its (communication) graph is strongly connected).*
A square matrix A is aperiodic if: $\exists m \geq 0, \ \forall i, j \mid (A^m)_{ij} > \varepsilon$.

DEFINITION 4.3 *For each circuit $\zeta = \{t_1, t_2, \cdots, t_j, t_{j+1} = t_1\}$, we define the average weight by:*

$$p(\zeta) = \frac{A_{t_1 t_j} \otimes \cdots \otimes A_{t_3 t_2} \otimes A_{t_2 t_1}}{j} ,$$

(here the division is the conventional one).

THEOREM 4.1 *There is a unique (non ε) eigenvalue, λ. It satisfies the relation*

$$\lambda = \max_{\zeta} p(\zeta) ,$$

where ζ describes all the circuits of (the communication graph of) A. We call also λ the **Lyapunov exponent** *of A.*

There might be several eigenvectors. An eigenvector has all its coordinates different from ε (due to the irreducibility assumption). A linear combination (in $\mathbb{R}_{Max}^{J}$) of eigenvectors is an eigenvector, i.e. if u_1 and u_2 are eigenvectors and $\alpha_1, \alpha_2 \in \mathbb{R}$, then $(\alpha_1 \otimes u_1) \oplus (\alpha_2 \otimes u_2)$ is also an eigenvector.

In particular, if u is an eigenvector and $\alpha \in \mathbb{R}$, then $\alpha \otimes u$ is also one. This was the motivation for the introduction of the projective space $\mathbb{PR}_{Max}^{J}$ (see definition 3.3). We will in general study the image by the canonical projection ($\pi : \mathbb{R}^J \to \mathbb{PR}^J$) of the set of eigenvectors (or periodic regimes) of a matrix.

Let us illustrate what the "linear combination of two vectors" means in $\mathbb{PR}_{Max}^{J}$. We consider examples of dimension 3. Let $u = (u_1, u_2, u_3)'$ and $v = (v_1, v_2, v_3)'$ be two vectors of $\mathbb{R}^3$. Let $\lambda, \mu \in \mathbb{R}$.

$$\pi(\lambda \otimes \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \oplus \mu \otimes \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}) = \pi(\begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \oplus (\mu - \lambda) \otimes \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}) .$$

The symbol $\bigwedge$ denotes the minimum of a finite set. We denote by $\bigwedge\bigvee$ the intermediate value of a set of three values. Let us assume for example that we have,

$$\bigwedge_{i=1,2,3}(u_i - v_i) = u_1 - v_1, \quad \bigwedge\bigvee_{i=1,2,3}(u_i - v_i) = u_2 - v_2,$$

$$\bigoplus_{i=1,2,3}(u_i - v_i) = u_3 - v_3 .$$

Depending on the value of $\alpha = \mu - \lambda$, there are four possible cases.

1. If $\alpha \leq \bigwedge(u_i - v_i)$, then $\pi(u \oplus \alpha v) = \pi(u)$.
2. If $\bigwedge(u_i - v_i) \leq \alpha \leq \bigwedge\bigvee(u_i - v_i)$, then $\pi(u \oplus \alpha v) = \pi(\alpha v_1, u_2, u_3)'$.
3. If $\bigwedge\bigvee(u_i - v_i) \leq \alpha \leq \bigoplus(u_i - v_i)$, then $\pi(u \oplus \alpha v) = \pi(\alpha v_1, \alpha v_2, u_3)'$.
4. If $\bigoplus(u_i - v_i) \leq \alpha$, then $\pi(u \oplus \alpha v) = \pi(v)$.

This particular example corresponds to the case of points C ($\pi(u)$) and A ($\pi(v)$) in Figure 4. The broken segment between C and A in Figure 4 is the set of linear combinations of the two points.
When two values are equal in $\{u_i - v_i,\ i = 1,2,3\}$, the picture is degenerate.

We are now ready to understand the form of sets of eigenvectors, knowing that linear combinations of eigenvectors are eigenvectors. We represent (in $\mathbb{R}^2 \simeq \mathbb{P}\mathbb{R}^3$) the image by π of the set of eigenvectors.

- $$M = \begin{pmatrix} 1 & e & e \\ e & 1 & e \\ e & e & 1 \end{pmatrix} .$$

The picture is exactly the same one as in Figure 4. The points A, B and C are the images by π of the columns of M which are eigenvectors (easy to check). The regular hexagon represented is the convex hull (in $\mathbb{R}_{Max}$) of these three points. It is the image by π of the set of eigenvectors of M.

- $$M = \begin{pmatrix} 1 & e & e \\ e & 1 & e \\ e & e & -1 \end{pmatrix} .$$

This case corresponds to Figure 5. The points A and B are (the image by π of) the two first columns of M. The broken segment between them is the set of eigenvectors of M. We obtain these eigenvectors as linear combination of A and B.

Let us recall some other definitions adapted to the $\mathbb{R}_{Max}$ algebra.

DEFINITION 4.4 *We normalize a matrix by dividing (in $\mathbb{R}_{Max}$ i.e. by subtracting in the conventional algebra) all its entries by its eigenvalue.*

A normalized matrix has e as eigenvalue. The eigenvectors are unchanged.

The eigenvalue of a matrix A gives the asymptotic growth rate of A^k/k (see Theorem 4.3 for a more precise statement). As a consequence, we will

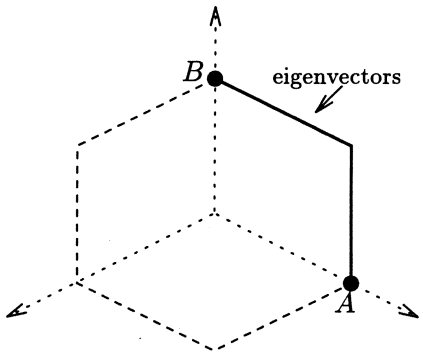


FIGURE 5. Set of eigenvectors

call problems related with the eigenvalue, first order problems. On the other hand eigenvectors are related with the problem of computing differences such as $A^{k+1}u - A^k u$. We call them second order problems (see [1] or [9]). In the rest of the paper, we will concentrate on second order results.
Eigenvectors and periodic regimes are invariant by a translation of all the entries of a matrix by the same real constant. In the rest of the paper, we will write the matrix we want to study in a positive form (i.e. with all terms $> e$) or in a normalized form depending on which one seems more convenient.

DEFINITION 4.5 *For a matrix A, with eigenvalue λ, we define:*

Critical circuit *A circuit ζ of A is said to be critical if its average weight is maximal, i.e. if $p(\zeta) = \lambda$.*

Critical graph *The critical graph consists of the nodes and arcs of A belonging to a critical circuit.*

Cyclicity *The cyclicity of a strongly connected graph (i.e. of an irreducible matrix) is the greatest common divisor of the lengths of all the circuits. The cyclicity of a general graph is the least common multiple of the cyclicities of its strongly connected subgraphs.*

The knowledge of the critical graph of a matrix accounts for much of its spectral behaviour. More precisely, to study the spectral behaviour of a matrix A, it is enough to know:

- The number of strongly connected subgraphs (**s.c.s.**) of its critical graph.
- The cyclicity of its critical graph.

In the following, a matrix whose critical graph is composed of j s.c.s. and whose cyclicity is k will be denoted **scs$j$-cyc$k$**.

The two fundamental theorems that we are going to present now justify the previous assertion. For a normalized matrix A of size J, we define:

$$A^+ = A \oplus A^2 \oplus \cdots \oplus A^J .$$

We check that $A^+ \oplus A^{J+1} = A^+$.

THEOREM 4.2 *Let A be a normalized matrix. Every eigenvector of A can be written as a linear combination of columns of A^+. More accurately, we have:*

1. *Column $A^+_{i.}$, i belonging to the critical graph, is an eigenvector.*
2. *$\pi(A^+_{i.})$ and $\pi(A^+_{j.})$ are different iff i and j belong to two different s.c.s. of the critical graph.*

Let p be the number of s.c.s. of the critical graph of A ($p \leq J$). The previous theorem states that there are p extremal eigenvectors. Then $p-1$ is the "dimension" of (the image of) the set of eigenvectors of A in $\mathbb{P}\mathbb{R}^J$. This set is polyhedral. The faces of this set are hyperplanes. These hyperplanes have a finite number of possible directions. We consider the natural basis of $(\mathbb{R}^J, +, \times)$:

$$(\mathbf{e_1}, \ldots, \mathbf{e_J}) \text{ with } \mathbf{e_i} = (0, \ldots, 0, 1, 0, \ldots, 0)' ,$$

the term 1 of $\mathbf{e_i}$ being in the i^{th} place. We choose $J-2$ vectors of this basis. We take their canonical projection. The hyperplane of $\mathbb{R}^{J-1} \simeq \mathbb{P}\mathbb{R}^J$ defined by these $J-2$ independent vectors is a possible direction for a face of the set of eigenvectors. We conclude that there are C_J^{J-2} possible directions for these hyperplanes.

For example in $\mathbb{P}\mathbb{R}^3$, there are $C_3^1 = 3$ possible directions for the lines delimiting the set of eigenvectors which are $\pi(1,0,0)', \pi(0,1,0)'$ and $\pi(0,0,1)'$. The lines will then be of the form

$$D : \pi\left(\begin{pmatrix} a \\ b \\ c \end{pmatrix} + \lambda \times \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}\right).$$

An example of this has already been given in Figure 4.

A corollary of Theorem 4.2 will be of particular use for us:
An irreducible matrix has a unique eigenvector (up to a multiplicative ($\otimes$) constant) iff its critical graph has a unique s.c.s.

In $\mathbb{R}^{Max}$, every irreducible matrix is cyclic in the sense of the next theorem.

THEOREM 4.3 *For an irreducible matrix A of size J and whose eigenvalue is λ, there exist integers d and M such that:*

$$\forall m \geq M, \quad A^{m+d} = \lambda^d \otimes A^m ,$$

($\lambda^d = \lambda^{\otimes d} = d \times \lambda$). Furthermore the smallest d verifying the property is equal to the cyclicity of the critical graph of A. From now on, we will call it the cyclicity of A.

The good interpretation is that there exists an initial transient regime for the powers of a matrix A. After the transient regime, the sequence $\{A^n\}$ becomes periodic (more rigorously, it is the sequence $\{\pi(A^n)\}$ which becomes periodic).

Sometimes, we will be interested only in the stationary regime, we will then consider directly A^M. On the other hand, we will sometimes consider the transient regime of a matrix.

If d is the cyclicity of A then A^d is of cyclicity one. A cyclicity greater than one will provide periodic regimes of period greater than one for the eigenvalue problem.

PROPOSITION 4.1 *An irreducible matrix has a unique eigenvector and no periodic regimes of period greater than one, iff its critical graph has a unique s.c.s. and its cyclicity is one, i.e. iff it is a* **scs1-cyc1** *matrix.*

Another easy consequence of Theorem 4.3 is the following where $d(.,.)$ is the projective distance and $u, v \in \mathbb{R}^J$.

$$\forall m \geq M,\ d(A^m u, A^m v) = d(A^M u, A^M v)\,.$$

4.2 Change of Basis

A matrix A of $\mathbb{R}_{Max}^{J\times J}$ can be considered as a linear (in a (Max,+) sense) operator on $\mathbb{R}_{Max}^{J}$. We want to have a formula of change of basis for the matrix associated with a given linear operator. We are only interested in permutations of the coordinates and translation of the origin.

DEFINITION 4.6 *A matrix P is called a matrix of permutation if there is one and only one term equal to e in each line and column, the other terms being equal to ε. Let $\mathcal{G}_J$ be the group of permutations of $\{1,\ldots,J\}$. We consider $\sigma \in \mathcal{G}_J$. The matrix of permutation associated with σ is P defined by:*

$$P_{\sigma(i),i} = e,\ P_{ji} = \varepsilon,\ \forall j \neq \sigma(i)\,.$$

LEMMA 4.1 *Let A be a $J \times J$ matrix and let $\hat{A}$ be the matrix associated with the same endomorphism in a new basis obtained from the original one by a permutation σ of the coordinates. Matrix P is the permutation matrix associated with σ and P^{-1} the one associated with σ^{-1}. We have*

$$\hat{A} = P^{-1} \otimes A \otimes P\,.$$

We consider a matrix A and we note $\tilde{A}$ the matrix associated with the same endomorphism in a new basis. We obtain the new basis from the original one by a translation of the origin of the projective space. Let $u \in \mathbb{R}^J$ be (a representative of) the new origin written in the old basis. Here is the formula of change of basis.

LEMMA 4.2 *Let A be a $J \times J$ matrix. Let u be a column vector of size J. We have (note the analogy with the conventional algebra):*

$$\tilde{A} = P^{-1} \otimes A \otimes P, \text{ where } P = \begin{pmatrix} u_1 & \varepsilon & \varepsilon \\ \varepsilon & \ddots & \varepsilon \\ \varepsilon & \varepsilon & u_J \end{pmatrix}.$$

Proof Let $v = (v_1, \cdots, v_J)'$ be a vector written in the old basis and let $\tilde{v} = (\tilde{v}_1, \cdots, \tilde{v}_J)'$ be this same vector in the new basis. We have $\tilde{v}_i + u_i = v_i$. We set $Av = w$ and $w = (w_1, \cdots, w_J)'$ and $(\tilde{w}_1, \cdots, \tilde{w}_J)'$ in the new and the old basis respectively.

$$\begin{aligned} (\tilde{A}\tilde{v})_i = (P^{-1} \otimes A \otimes P\tilde{v})_i &= (P^{-1} \otimes Av)_i \\ &= (P^{-1}w)_i = \tilde{w}_i \end{aligned}$$

■

An illustration of such a change of origin is provided by Figure 17. It might be interesting to get another intuition on what a change of origin means. We present now an interpretation suggested by the modelling of Stochastic Event Graphs. Let us consider the communication graph associated with a positive and irreducible matrix $A \in \mathbb{R}_{Max}^{J \times J}$. We consider that there is a clock associated with each node of A. Let u be a vector of $\mathbb{R}^J$. We interpret u_i as a date of occurrence of a first event at node i. Then $(Au)_j$ is interpreted as the date of occurrence of the second event at node j. In this framework, a "change of origin" is just a change of the origin of time for some or all of the daters. It does not modify of course the evolution of the system.

Let us prove a very useful lemma. This lemma together with the previous one enables us to determine in which cases critical terms are greater than non-critical ones.

LEMMA 4.3 *We consider a matrix A, irreducible, of size J, and u an eigenvector of A. Let P be the matrix of change of the origin associated with u. We define $\tilde{A} = P^{-1} \otimes A \otimes P$. We have the following property:*

$$\forall i, j \in 1, \cdots, J, \quad \tilde{A}_{ij} \leq \lambda,$$

and $\forall p, q$ such that (p, q) belongs to the critical graph, we have $\tilde{A}_{qp} = \lambda$, where λ is the Lyapunov exponent of A.

Proof We set $\mathbf{e} = (e, \ldots, e)'$.

$$\tilde{A}\mathbf{e} = \left(\bigoplus_k \tilde{A}_{1k}, \ldots, \bigoplus_k \tilde{A}_{Jk} \right).$$

But using the fact that $\mathbf{e}$ is an eigenvector of $\tilde{A}$ ($P^{-1}AP\mathbf{e} = P^{-1}Au = P^{-1}\lambda u = \lambda\mathbf{e}$) and the definition of the Lyapunov exponent, we get:

$$\tilde{A}\mathbf{e} = (\lambda, \cdots, \lambda)' .$$

Then $\forall i$, $\bigoplus_k \tilde{A}_{ik} = \lambda$, which proves the first part of the lemma. Let us suppose there exist p, q such that (p, q) belongs to the critical graph and $\tilde{A}_{qp} < \lambda$. There is a critical circuit involving the arc (p, q). Using the first part of the theorem and $\tilde{A}_{qp} < \lambda$, we conclude that the mean weight of this critical circuit is strictly smaller than λ, which is a contradiction. ∎

4.3 Spectral Theory in Dimension 3

We are now ready to have a closer look at aperiodic matrices of size 3. We are going to present an exhaustive inventory of the possible spectral behaviours. Using theorems 4.2 and 4.3, we show that there are only six possible cases, which can be sorted in four categories.

- scs1-cyc1.
- scs3-cyc1 and scs1-cyc3.
- scs2-cyc1 and scs1-cyc2.
- scs2-cyc2.

We are going to study them one after the other in the simplest case when all non-critical terms are equal. Then we will observe that the general behaviour is stable under small perturbations of non-critical terms. We will show precisely how these perturbations modify the behaviour. By this way, we will have described all possible aperiodic matrices of size 3.

In order for the reader to be convinced that all the cases are treated, we propose a classifying algorithm which, given a specific matrix, associates to it a paragraph and one or several figures of the paper.

We consider a matrix $A \in \mathbb{R}_{Max}^{J \times J}$.

Algorithm

1. Check if A is irreducible and aperiodic.
2. Normalize matrix A.
3. Find an eigenvector of A.
4. Write A in a new basis.
5. Determine the critical graph of A.
6. Compute the projective size of A.
7. Check non critical terms of A. Final classification.

Let us detail the different stages.

STAGE 1 *Check if A is irreducible and aperiodic.*

In order to do this, an easy way is to consider $\tilde{A}$ the boolean matrix associated with A. It is defined in the following way: $\tilde{A}_{ij} = \varepsilon$ if $A_{ij} = \varepsilon$, $\tilde{A}_{ij} = e$ if $A_{ij} \geq \varepsilon$. It is sufficient to consider $\tilde{A}, \tilde{A}^2, \ldots, \tilde{A}^J$ to conclude.
Matrix A is irreducible if and only if $\tilde{A} \oplus \tilde{A}^2 \oplus \cdots \oplus \tilde{A}^J = E$, where E is defined by $E_{ij} = e, \forall i, j$.
Matrix A is aperiodic if and only if $\tilde{A} = E$ or $\forall k = 2, \ldots, J$ $\tilde{A}^k \neq \tilde{A}$.

STAGE 2 *Normalize matrix A.*

Compute the smallest integer, denoted M, such that:

$$\exists d \in \mathbb{N},\ \exists \lambda \in \mathbb{R} \mid \forall m \geq M,\ A^{m+d} = \lambda^d \otimes A^m .$$

The eigenvalue of A is λ. Normalize matrix A. For simplicity, we will keep the original notation, i.e. $A := A - \lambda$.

STAGE 3 *Find an eigenvector of A.*

Here is a general method, used for example in [5], to compute an eigenvector of A. An alternative algorithm, valid for some kind of matrices only, is proposed in Section 5, Example 1.

- Consider an i_0 such that $\exists k \mid (A^k)_{i_0 i_0} = e$. By Theorem 4.1, this condition is verified if and only if i_0 belongs to the critical graph of A.
- Compute:

$$u = \bigoplus_{k \geq 0} A^k \delta_{i_0}, \quad \text{where } (\delta_{i_0})_{i_o} = e,\ (\delta_{i_0})_j = \varepsilon,\ j \neq i_0 .$$

  Then u is an eigenvector of A. In the previous formula, it is enough to compute the powers of A until J.

Let us prove rapidly this last assertion. We denote by I the matrix defined by $I_{ii} = e$, $I_{ij} = \varepsilon$, $i \neq j$. We have:

$$u = \bigoplus_{k \geq 0} A^k \delta_{i_0} = I\delta_{i_0} \oplus \bigoplus_{k \geq 1} A^k \delta_{i_0} = \delta_{i_0} \oplus Au .$$

We deduce that $Au \geq u$. Assume we have $Au \neq u$, then we must have $(Au)_{i_0} > u_{i_0}$ as we clearly have $(Au)_j = u_j, \forall j \neq i_0$. But as we have $e \oplus (Au)_{i_0} = u_{i_0}$ we see that it is also contradictory to suppose $(Au)_{i_0} > u_{i_0}$. ■

STAGE 4 *Write A in a new basis.*

Consider the linear operator associated with A. Consider a new basis obtained from the original one by a translation of the origin. The new origin is the eigenvector of A calculated in the previous stage. Write the operator associated with A in this new basis. For simplicity, we keep the notation A for the operator in the new basis. By Lemma 4.2, we have:

$$A := P^{-1}AP, \quad \text{where } P = \begin{pmatrix} u_1 & \varepsilon & \varepsilon \\ \varepsilon & \ddots & \varepsilon \\ \varepsilon & \varepsilon & u_J \end{pmatrix} .$$

By Lemma 4.3, all critical terms of A are now equal to e and all non-critical terms are less than or equal to e. We recall that A_{ij} is a critical term if the arc (i,j) belongs to the critical graph.

STAGE 5 *Determine the critical graph of A.*

Determine the critical graph. It suffices to draw the (communication) graph of terms equal to e in A and to keep only the circuits of this graph.
Compute the number of strongly connected subgraphs (j) and the cyclicity (k) of the critical graph. The paragraph corresponding to the general spectral behaviour of A is **scsj-cyck**.

STAGE 6 *Compute the projective size of A.*

Consider A^M the stationary version of A as defined previously. We define critical columns (resp. lines) as columns (resp. lines) of A^M containing a critical term. We denote $C = \{(i,j) \mid (i,j)$ belongs to a critical line or a critical column $\}$. Set:

$$\alpha = \bigwedge_{(i,j)\in C} A^M_{ij} .$$

We call α the projective size[1] of A. If $\alpha = 1$, we are exactly in the frame of the examples and of the figures considered below. If $\alpha \neq 1$, the correct figure is obtained from the ones drawn below by an homothetic transformation of center $\mathbf{e} = (e,e,e)'$ and of ratio α.

STAGE 7 *Check non critical terms of A. Final classification.*

Consider the couples $(i,j) \in C$ which do not belong to the critical graph. If they are all equal to α, then the figures corresponding to matrix A are given in the first table. If these couples are not all equal to α, the figures get modified. One has now to report to the figures of table 2.

Remark In the scs2-cyc1 and scs1-cyc2 cases, the figures correspond to the situation where the critical columns are 1 and 2. If this is not the case of matrix A, consider a new basis obtained from the original one by a permutation of the coordinates (see Lemma 4.1). In the same way, in the scs2-cyc2 case, when the cycle of length 2 is not over coordinates (1,2), write A in a new basis obtained by permutation of the coordinates.

Table **1.**

Type of A	*Figure n°*
scs3-cyc1	6
scs1-cyc3	8
scs2-cyc1	10
scs1-cyc2	10
scs2-cyc2	12

, Table **2.**

Type of A	*Figure n°*
scs3-cyc1	7
scs1-cyc3	9
scs2-cyc1	11
scs1-cyc2	11
scs2-cyc2	

.

[1] In the scs3-cyc1 and scs2-cyc1 cases, it is exactly equal to the projective radius of A. We recall that the projective diameter of A is defined by $\mathrm{D}(A) = \sup_{u,v\in \mathbf{R}^J} d(Au, Av)$.

Remark If A is scs1-cyc1, there is no figure as the spectral behaviour is trivial.

Remark This algorithm is the easiest to work with when dealing with matrices of small dimensions. It is not the best one in terms of complexity. A better algorithm is obtained by using Karp's algorithm (see [4]) in order to compute the eigenvalue in stage 2. One can then obtain all the eigenvectors by computing the Kleene's star of the normalized matrix (see theorem 4.2). This can be done by using Gauss algorithm (see for example [6]). Both Karp's and Gauss algorithms have a complexity $O(J^3)$, where J is the size of the matrix.

Let us consider the six possible spectral behaviours one after the other. For each case, we are going to draw the set of eigenvectors and periodic regimes, in $\mathbb{P}\mathbb{R}^3 \simeq \mathbb{R}^2$.

We will also represent the domains of attraction of the different eigenvectors and periodic regimes. For a matrix A, we call domain of attraction of an eigenvector (or of a periodic regime) the set of initial conditions $\{x_0\}$ such that $\pi(A^n x_0)$ converges to that eigenvector (or periodic regime). By Theorem 4.3, this convergence occurs in finite time.

- **scs1-cyc1**

Let A be a scs1-cyc1 matrix. We denote by v the unique eigenvector of A. Proposition 4.1 together with theorem 4.3 gives:

$$\forall u_0 \in \mathbb{R}^J, \ \pi(A^m u_0) \xrightarrow{m} \pi(v) .$$

The convergence occurs in finite time (Theorem 4.3). The domain of attraction of $\pi(v)$ is $\mathbb{P}\mathbb{R}^J$ and the initial condition u_0 is forgotten. This case is of special importance for stochastic models (see [9]).

- **scs3-cyc1** and **scs1-cyc3**

If A is a scs1-cyc3 matrix, then A^3 is a scs3-cyc1 matrix (but the converse is false!). For example,

$$A = \begin{pmatrix} . & . & e \\ e & . & . \\ . & e & . \end{pmatrix}, \ B = A^3 = \begin{pmatrix} e & . & . \\ . & e & . \\ . & . & e \end{pmatrix},$$

where (.) stands for -1. We consider first the scs3-cyc1 case.

There are three independent eigenvectors and no periodic regime of period greater than one (Theorem 4.2). Let us consider more specifically the matrix B. B is a normalized matrix and we check that it is stationary (i.e. $B^2 = B$). We have represented in Figure 6 the set of eigenvectors and the domains of attraction. Theorem 4.2 helps us understand this picture. Here we have $B^+ = B$. Then the three columns B_1, B_2 and B_3 of B are the extremal

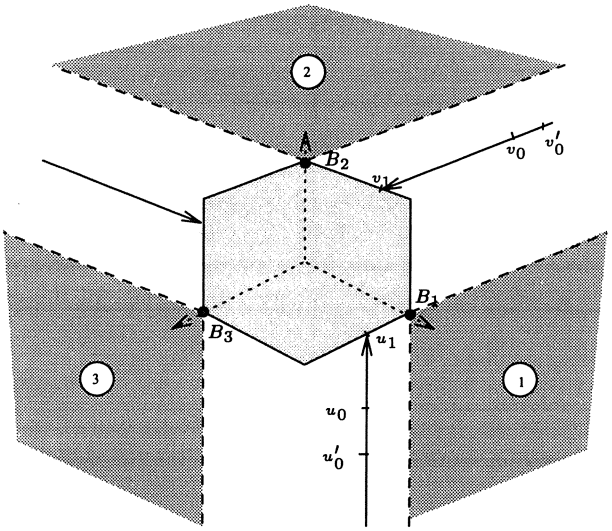


FIGURE 6. scs3-cyc1, domains of attraction.

eigenvectors. These extremal eigenvectors (or more precisely their image by π) are represented by a black dot ($\bullet$). The set of eigenvectors is the $\mathbb{R}_{Max}$ convex hull of these three eigenvectors. It is filled in light gray.
If the initial condition x_0 is in the dark gray zone number i, then the limit value (of $\pi(B^n x_0)$) is $\pi(B_i)$. If the initial condition is in one of the white strips, then the limit value is the nearest point for the projective distance (and of course this limit is attained in one step as $B^2 = B$). For example, for initial conditions u_0 or u'_0 (resp. v_0, v'_0) the limit value is u_1 (resp. v_1).

We will now consider what happens if we modify the non-critical terms of the matrix B. We consider three different examples to illustrate it.

$$C = \begin{pmatrix} e & . & . \\ -0.5 & e & . \\ . & . & e \end{pmatrix}, \; D = \begin{pmatrix} e & -0.6 & . \\ . & e & 0.6 \\ . & . & e \end{pmatrix},$$

$$E = \begin{pmatrix} e & . & -0.5 \\ -0.2 & e & . \\ -0.2 & -0.5 & e \end{pmatrix},$$

where $(.) = -1$. We represent the set of eigenvectors and the limits between domains of attraction.

We can represent these sets very rapidly, using the procedure described at the end of Section 3. Let us consider the matrix C first. We represent the three columns of C, $\pi(C_1)$, $\pi(C_2)$ and $\pi(C_3)$. The convex hull of these three points is the set of eigenvectors of C (note that $C = C^2$). For the matrix E, the interpretation is the same.

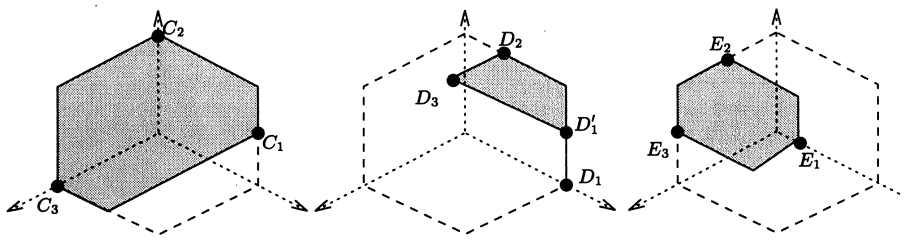


FIGURE 7. scs3-cyc1, three possible forms for the set of eigenvectors.

We consider now the matrix D. The difference with the two previous examples is that D is not stationary. The convex hull of the columns of D, $\pi(D_1)$, $\pi(D_2)$ and $\pi(D_3)$, is the image of D ($D(\mathbb{R}^3)$). It is different from the set of eigenvectors which is, here, the interior of this convex hull. Another way to obtain the set of eigenvectors is to consider the stationary version of D (i.e. D^2, as we have $D^3 = 1 \otimes D^2$). The set of eigenvectors is the convex hull of the columns of D^2, $\pi(D_1')$, $\pi(D_2)$ and $\pi(D_3)$.

Now we consider the case of scs1-cyc3 matrices. There is only one eigenvector but there are periodic regimes of period 3. The set of periodic regimes of period 3 of a scs1-cyc3 matrix M is equal to the set of eigenvectors of M^3. Let us consider more specifically the matrix A defined previously.

$$A = \begin{pmatrix} . & . & e \\ e & . & . \\ . & e & . \end{pmatrix}, \ (.) = -1 .$$

The previous remark provides us with the set Π of periodic regimes of A. To go further, we want to characterize, given an initial condition u in the hexagon Π, the periodic regime $\{u, Au, A^2u\}$.

It is easy to check that the unique eigenvector of A is $\mathbf{e} = (e, e, e)'$. We consider $u \in \Pi$, $u \neq \mathbf{e}$. Theorem 4.3 shows us that $\{u, Au, A^2u\}$ is a periodic regime. It implies that $A^3u = u$ and $d(A^3u, \mathbf{e}) = d(u, \mathbf{e})$. By Proposition 3.1, we have

$$d(A^3u, \mathbf{e}) \leq d(A^2u, \mathbf{e}) \leq d(Au, \mathbf{e}) \leq d(u, \mathbf{e}) .$$

We conclude that:

$$d(A^2u, \mathbf{e}) = d(Au, \mathbf{e}) = d(u, \mathbf{e}) .$$

The points of a periodic regime are at a constant distance (for the projective distance) of the unique eigenvector $\mathbf{e}$. The symmetry does the trick as the figure constituted by the three points $\{u, Au, A^2u\}$ must be invariant by a permutation of the three projective axes. The direction of rotation depends on

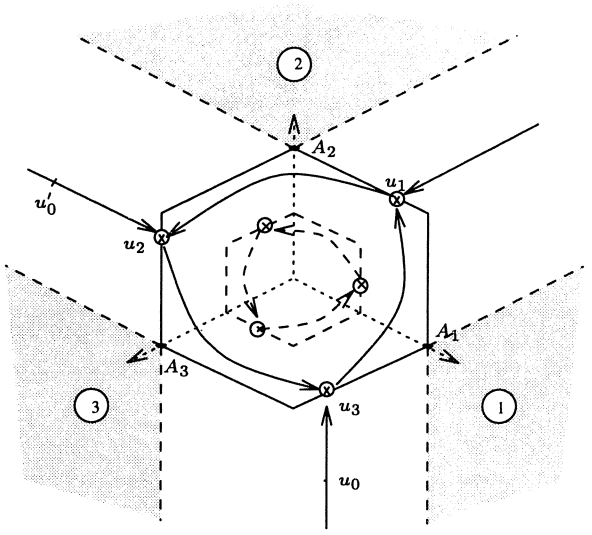


FIGURE 8. scs1-cyc3, periodic regimes.

---

the critical cycle, whether it is $(1,2,3)$ or $(1,3,2)$. For example A and A^2 have opposite directions of rotation.

In Figure 8, we have represented two periodic regimes of A (the columns of A, $\pi(A_1), \pi(A_2)$, $\pi(A_3)$ constitute a third one). The direction of rotation is counter-clockwise. We have also represented the domains of attraction. If the initial condition is in one of the gray zones then the stationary periodic regime is $\pi(A_1), \pi(A_2)$ or $\pi(A_3)$. If the initial condition is in one of the white strips, the limit regime consists of three points on the boundary of the hexagon. We have represented an example. It corresponds to initial conditions along one of the three large arrows. For example for an initial condition u_0 or u'_0, the limit regime is $\{u_1, u_2, u_3\}$. More precisely, we have:

$$\pi(Au_0) = \pi(u_1),\ \pi(A^2u_0) = \pi(u_2),\ \pi(A^3u_0) = \pi(u_3),\ \pi(A^4u_0) = \pi(u_1), \dots ,$$

$$\pi(Au'_0) = \pi(u_3),\ \pi(A^2u'_0) = \pi(u_1), \dots .$$

If the initial condition u belongs to Π, the stationary periodic regime is $\{u, Au, A^2u\}$ of course. We have also drawn an example of such a regime.

What happens if we perturb non-critical terms? To describe it, it will be useful to define the notion of subdiagonals.

DEFINITION 4.7 *Let M be a matrix of size J. We call i^{th} subdiagonal of M the terms* $\{M_{i1}, M_{i+1,2}, \dots, M_{i+J-i,1+J-i}, M_{1,2+J-i}, \dots, M_{i-1,J}\} = \{M_{i-1+k,k\ [J]}, \forall k\}$.

For example, the first subdiagonal is the diagonal of the matrix! For the matrix A above, the critical subdiagonal is the second one. If we perturb a non-critical term (i.e. a term of the first or third subdiagonal), after a transient regime, the whole subdiagonal will be equal to this term. Let us consider an example.

$$A' = \begin{pmatrix} a & b_1 & e \\ e & . & . \\ b_2 & e & . \end{pmatrix} \longrightarrow (A')^4 = \begin{pmatrix} a & b & e \\ e & a & b \\ b & e & a \end{pmatrix},$$

$$(A')^5 = \begin{pmatrix} b & e & a \\ a & b & e \\ e & a & b \end{pmatrix}, \ldots,$$

with $(.) = -1$, $-1 < a, b_1, b_2 < e$, $b = b_1 \oplus b_2$[1]. This provides us with specific pictures for the sets of periodic regimes. When we increase continuously a non-critical term, this set evolves in the same manner as the diaphragm of a camera. Let us illustrate it in Figure 9.

$$F = \begin{pmatrix} -0.8 & . & e \\ e & -0.8 & . \\ . & e & -0.8 \end{pmatrix}, \; G = \begin{pmatrix} -0.5 & . & e \\ e & -0.5 & . \\ . & e & -0.5 \end{pmatrix},$$

$$H = \begin{pmatrix} -0.2 & . & e \\ e & -0.2 & . \\ . & e & -0.2 \end{pmatrix}, \; (.) = -1 .$$

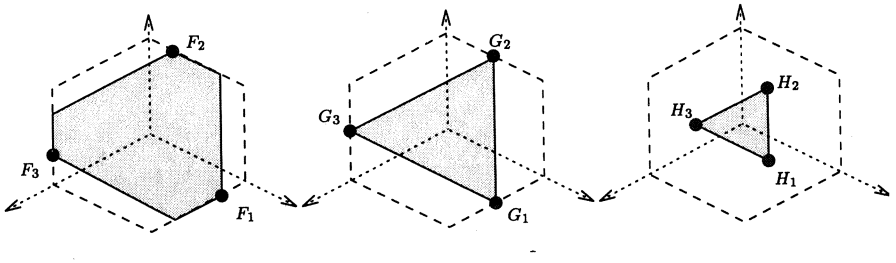


FIGURE 9. scs1-cyc3, three possible forms for the set of eigenvectors.

When the terms of the diagonal become equal to e, then we have a scs1-cyc1 matrix with $\mathbf{e} = (e, \ldots, e)'$ as unique eigenvector. If the terms of the diagonal are greater than e, then we get a scs3-cyc1 matrix where we find the same kind of pictures as in Figure 6, sets which have now to be interpreted in terms of eigenvectors.

Remark In the cases we have been dealing with so far, domains of attraction had a very easy algebraic characterization. In fact for an initial condition

[1] The size of matrix A is here $inf\ (a, b)$.

u the limit value was the "nearest" (for the projective distance) eigenvector or periodic regime. This is a general result. It is a consequence of the synchronization phenomena occuring in $\mathbb{R}_{Max}$. However we will see, in the scs1-cyc2 case for example, that the "nearest" eigenvector is not always unique which complicates the description of domains of attraction.

- **scs2 − cyc1 and scs1 − cyc2.**

In the same way as previously, if A is a scs1-cyc2 matrix then A^2 is a scs2-cyc1 matrix, the converse being false. For example,

$$A = \begin{pmatrix} . & e & . \\ e & . & . \\ . & . & -2 \end{pmatrix}, \; B = A^2 = \begin{pmatrix} e & . & . \\ . & e & . \\ . & . & -2 \end{pmatrix}, \; (.) = -1 .$$

Let us consider first the scs2-cyc1 case and more precisely the matrix B. The general results of spectral theory tell us that there are two extremal eigenvectors (the first two columns of B as $B^+ = B$) and no periodic regime of period greater than 1. We have already represented the set of eigenvectors of B, in Figure 5. We will represent it again together with the domains of attraction of the different eigenvectors in Figure 10.

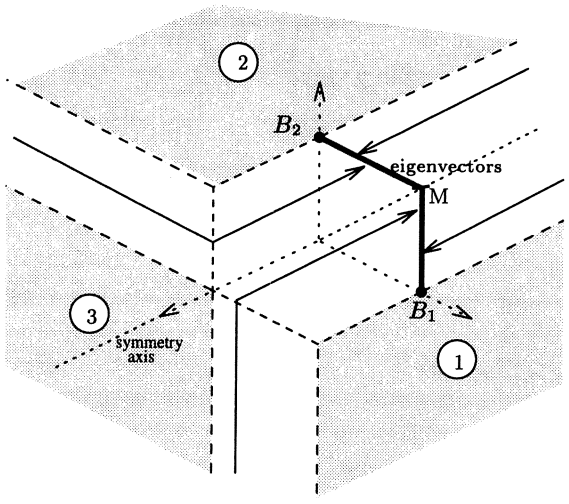


FIGURE 10. scs2-cyc1 (or scs1-cyc2), domains of attraction.

There is a symmetry axis for the whole figure (corresponding to the fact that matrix B is unchanged by a permutation of the first two columns). The extremal eigenvectors are symbolized by a dot, •. In opposition with the scs3-cyc1 case, no eigenvector has a domain of attraction restricted to itself. If the initial condition x_0 is in the gray zones 1 or 2, the limit value of $\pi(B^k x_0)$ will be B_1 or B_2 respectively. If it is in zone 3, then the limit value will be M.

When the initial condition is in one of the white strips, the limit value is given by the arrow.

The picture remains the same for the matrix A which is scs1-cyc2. There is only one eigenvector which is M. The bold "line" between B_1 and B_2 is the set of periodic regime of period 2. Two points of this set belong to the same periodic regime if they are "symmetric" with respect to M. For an initial condition in zone 3, the limit regime is the eigenvector M. For an initial condition in zones 1 or 2, the limit regime is $\{B_1, B_2\}$ and so on.

Remark In general, for an initial condition in zone 3, the limit regime is $\lim_k \pi(B^k\delta_3)$ where $\delta_3 = (\varepsilon, \varepsilon, e)'$. It is the third column of the matrix in its stationary version but it is not always the eigenvector of the matrix. However, by using theorem 4.3, it is possible to prove that

$$\exists \alpha, K \; s.t. \; \forall k > K \; \pi(B^k\delta_3) = \pi(B^k\delta_1) \oplus \alpha \otimes \pi(B^k\delta_2) \, .$$

Some of the possible cases are now investigated.

We want to analyze what happens if we modify non-critical terms. We have to distinguish between modifications of terms belonging to critical columns (columns 1 and 2 here) and of terms belonging to non-critical columns. If we modify a term belonging to a critical column, the set of eigenvectors (obtained as the convex hull of critical columns) will also be modified. On the other hand, it is possible that a modification of a term of the non-critical column does not affect the set of eigenvectors but only the domains of attraction. Let us illustrate this idea.

$$C = \begin{pmatrix} e & . & -0.5 \\ . & e & . \\ . & . & -2 \end{pmatrix}, \; D = \begin{pmatrix} e & . & 0.5 \\ . & e & . \\ . & . & -2 \end{pmatrix}, \; (.) = -1 \, .$$

For matrix C, the set of eigenvectors is not modified, but the domains of attraction are. The picture of Figure 11 has to be interpreted in the same way as previously. The gray zones 1 and 2 are the domains of attraction of C_1 and C_2 respectively. If the initial condition u_0 is in zone 3, the limit value of $\pi(B^k u_0)$ will be M.

For matrix D, the domains of attraction and the set of eigenvectors are modified. In fact, the stationary regime of D is:

$$D^2 = \begin{pmatrix} e & -0.5 & 0.5 \\ . & e & -0.5 \\ . & . & -0.5 \end{pmatrix}, \; (.) = -1 \, .$$

For matrix D^2, a term of a critical column has been modified. It is reflected by a corresponding modification of the set of eigenvectors. The points represented, D_1 and D_2' are the critical columns of matrix D^2. In this example, zones 1 and 3 have melted. They constitute the domain of attraction of D_1.

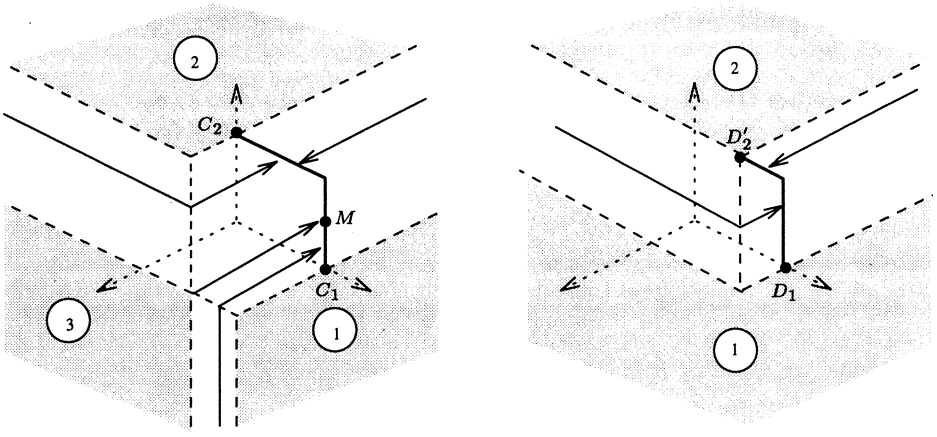


FIGURE 11. scs2-cyc1, other examples, matrices C and D.

- **scs2-cyc2**

The canonical example of such a matrix is:

$$A = \begin{pmatrix} . & e & . \\ e & . & . \\ . & . & e \end{pmatrix}, \ (.) = -1 .$$

There are two extremal eigenvectors and also periodic regimes of period 2. If a matrix N is scs2-cyc2 then the matrix N^2 is scs3-cyc1. Then to find the set of eigenvectors and periodic regimes of a scs2-cyc2 matrix N, one only has to determine the set of eigenvectors of N^2 (see paragraph **scs3-cyc1** and **scs1-cyc3**).

Let us represent graphically eigenvectors, periodic regimes of period 2 and domains of attraction of matrix A in Figure 12.

There is a symmetry axis for the whole figure (matrix A is unchanged by a permutation of the first two coordinates). The set of eigenvectors (the interval $[M, A_3]$) splits the set of periodic regimes in two equal parts. The two points of a periodic regime of period 2 are symmetric with respect to the set of eigenvectors. The analysis of domains of attraction is analog to the one of cases scs3-cyc1 and scs1-cyc3. If the initial condition belongs to the zones 1, 2 or 3, the limit value will be either the periodic regime $\{A_1, A_2\}$ or A_3. If the initial condition belongs to one of the three white strips, the limit regime is a periodic regime of period 2, corresponding to the "nearest" point on the hexagon and its symmetrical point.

We can also modify non-critical terms. We will not present examples but the behaviour is very close to the one observed in the previous cases.

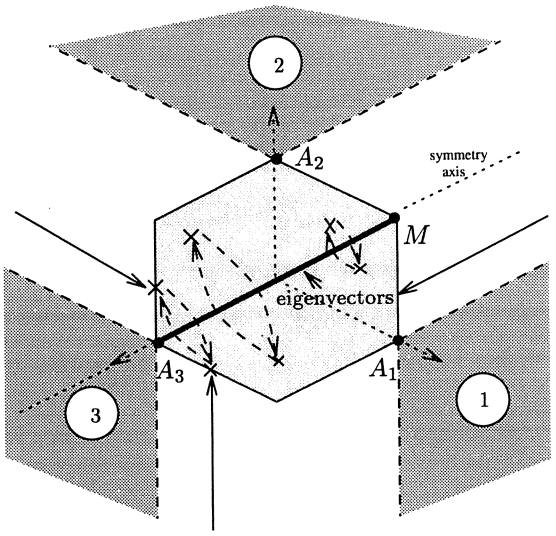


FIGURE 12. scs2-cyc2, domains of attraction.

4.4 Transient regimes

We will now take a closer look at transient regimes of matrices. The matrices we have been considering so far were chosen in order to be stationary or at least to have a very short transient regime. To emphasize the transient behaviour, we will, on the other hand, consider matrices with long transient regimes. The length of the transient regime is closely related to the "second eigenvalue" of the matrix, i.e the second largest circuit weight (see [4]).

First of all, one has to remark that a matrix can have an arbitrarily long transient regime. Let us take an example.

$$M = \begin{pmatrix} e & -1 \\ -1 & -\eta \end{pmatrix},\ 0 < \eta \ll 1,\ M^2 = \begin{pmatrix} e & -1 \\ -1 & -2 \times \eta \end{pmatrix},$$

$$M^k = \begin{pmatrix} e & -1 \\ -1 & -k \times \eta \end{pmatrix}, k < [\frac{2}{\eta}] + 1, M^k = \begin{pmatrix} e & -1 \\ -1 & -2 \end{pmatrix}, k \geq [\frac{2}{\eta}] + 1.$$

The length of the transient regime is thus $[\frac{2}{\eta}]$. The matrix M is scs1-cyc1, its unique eigenvector is $E = (e, -1)'$. As we have seen previously, it implies that $\forall u \in \mathbb{R}^J$, $\lim_k \pi(M^k u) = \pi(e, -1)'$. Let us consider the initial condition $u = (e, 3)'$. We have $\pi(Mu) = \pi(e, 1 - \eta)'$, $\pi(M^2 u) = \pi(e, 1 - 2 \times \eta), \ldots$. We have represented the sequence $\{\pi(M^k u)\}$ in the projective space $\mathbb{P}\mathbb{R}^2$. We have also represented the same sequence for three other initial conditions.

We are now going to present analog figures corresponding to matrices of size 3.

First of all we consider the example of scs1-cyc1 matrices.

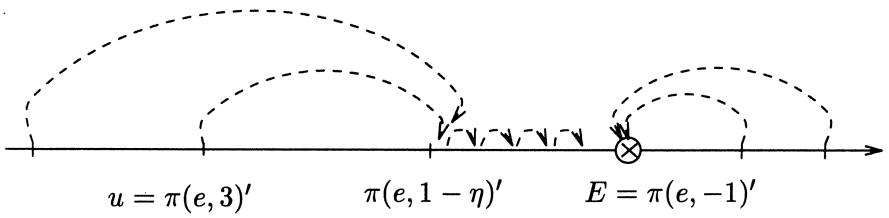


FIGURE 13. Dimension 2, transient regimes.

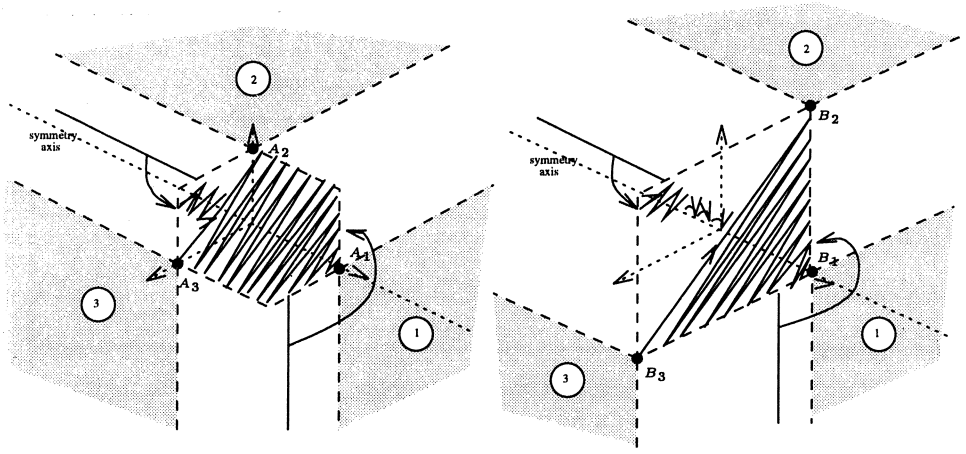


FIGURE 14. Dimension 3, scs2-cyc2, transient regimes.

$$A = \begin{pmatrix} e & . & . \\ . & . & -\eta \\ . & -\eta & . \end{pmatrix}, \quad B = \begin{pmatrix} e & . & . \\ . & -2 & -\eta \\ . & -\eta & -2 \end{pmatrix}, \quad 0 < \eta \ll 1, \ (.) = -1 .$$

Both matrices have the same stationary regime, given by the matrix:

$$\lim_k A^k = \lim_k B^k = \begin{pmatrix} e & . & . \\ . & -2 & -2 \\ . & -2 & -2 \end{pmatrix}, \ (.) = -1 .$$

Matrix A is obtained by a small perturbation of the matrix presented in paragraph **scs2-cyc2** (Figure 12). The transient behaviour reflects it, as the figure we obtain is very close to Figure 12. As a comparison, we have also represented the matrix B whose behaviour is asymptotically identical.

Let us comment on the figure corresponding to A a little further. There is a symmetry axis for the whole picture. The three points A_1, A_2 and A_3 are the projections of the columns of matrix A. If the initial condition is in zone 1, there is convergence in one step to $A_1 = \pi(e,-1,-1)'$, the unique eigenvector.

If the initial condition is in zone 2 (resp. 3), we have $\pi(Ax_0) = A_3$ (resp. $\pi(Ax_0) = A_2$). We have represented the whole sequence $\{\pi(A^k x_0)\}$ for an initial condition $x_0 = A_3$. For an initial condition in one of the three white strips, for example let us consider u_0 (or u_0'), then $\pi(Au_0)$ (or $\pi(u_0')$) is the point pointed by the arrow in the picture (it is the symmetric of the "nearest" point on the set $Im(A)$). For initial condition u_0', we have also drawn the beginning of the sequence $\{\pi(A^n u_0')\}$.

For matrix B, the set of periodic regimes is the same one as A. But the domains of attraction are quite modified. It emphasizes the possible influence of transient regimes, especially in stochastic models. Here we have drawn the sequences $\{\pi(A^n u_0)\}$ (or part of them) for several different initial conditions. One of them is in zone 2, another one in the white strip between zones 2 and 3 and the last one is on the symmetry axis.

We consider now the transient regime of a scs2-cyc1 matrix.

$$C = \begin{pmatrix} e & . & . \\ . & e & . \\ . & . & -\eta \end{pmatrix}, \; 0 < \eta \ll 1, \; (.) = -1 \, .$$

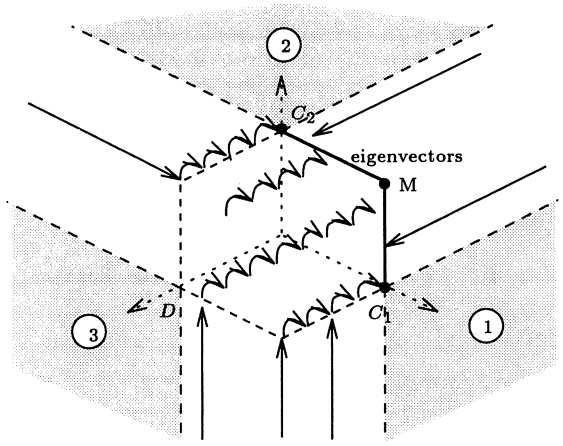


FIGURE 15. Dimension 3, scs2-cyc1, transient regimes.

The stationary regime of C is the canonical example of the **scs2-cyc1** paragraph (i.e. the same matrix with $-\eta = -2$). We can also view C as a small perturbation of the canonical example of **scs3-cyc1** matrix (i.e the same matrix with $\eta = 0$). The figure reflects these remarks.
The points C_1 and C_2 are the projection of the first two columns of the matrix. The point D is $\pi(e, e, 1-\eta)'$. If the initial condition is in zone 1 (resp. 2) we have convergence to C_1 (resp. C_2) in one step. If it is in the strip between zone 1 and 2, convergence occurs in one step according to the arrows. The hexagon

represented in dotted line is $Im(C) = C(\mathbb{R}^3)$. We have represented the whole sequence $\{\pi(C^k x_0)\}$ for several initial conditions.

5 APPLICATIONS

We are now going to propose examples where the graphical representation in the projective space appears to be useful in understanding some (Max,+) phenomena.

5.1 A projectively infinite semigroup of matrices.

We consider a finite number of matrices $A_1, \ldots, A_k \in \mathbb{R}_{Max}^{J\times J}$. We denote respectively by $< A_1, \ldots, A_k >$ and $\pi< A_1, \ldots, A_k >$ the semigroup generated by $A_1, \ldots, A_k$ and its projection.

$$< A_1, \ldots, A_k >= \{(A_{u_N} \cdots A_{u_2} A_{u_1}), u_1, \ldots, u_N \in \{1, \ldots, k\}, N \ finite\},$$

$$\pi< A_1, \ldots, A_k >= \{\pi(A_{u_N} \cdots A_{u_2} A_{u_1}),\ u_1, \ldots, u_N \in \{1, \ldots, k\}, N \ finite\},$$

where π is here the canonical projection of $\mathbb{R}_{Max}^{J\times J}$ into $\mathbb{P}\mathbb{R}_{Max}^{J\times J}$. The problem we are interested in is the finiteness of $\pi< A_1, \ldots, A_k >$. It is in fact a version of the Burnside problem in the special case of the (Max,+) algebra (see [7]).

Let us consider the projective semigroup generated by a single and irreducible matrix $\pi< A >= \{\pi(A), \pi(A^2), \ldots\}$. Theorem 4.3 tells us that $\pi< A >$ is finite.

Remark It is the finiteness of the projective semigroup and not the finiteness of the semigroup which is interesting. Indeed any irreducible matrix A with an eigenvalue different from e is such that $\#\{< A >\} \equiv \text{Card}\{< A >\}$ is infinite.

The next theorem was proved in [7] in a slightly stronger version.

THEOREM 5.1 *Let $A_1, \ldots, A_k \in \mathbb{Q}_{Max}^{J\times J}$. We assume that:*

$$\forall u \in \{1, \ldots, k\}, \forall (i,j),\ (A_u)_{ij} > \varepsilon\ .$$

Then the projective semigroup $\pi< A_1, \ldots, A_k >$ is finite.

This theorem does not extend to the case of matrices with non rational entries. Nice counter-examples can be found using the graphical representation in the projective space.
We consider the semigroup generated by the matrices:

$$A_1 = \begin{pmatrix} -\eta_1 & . & . \\ . & e & . \\ . & . & e \end{pmatrix}, A_2 = \begin{pmatrix} e & . & . \\ . & -\eta_2 & . \\ . & . & e \end{pmatrix}, A_3 = \begin{pmatrix} e & . & . \\ . & e & . \\ . & . & -\eta_3 \end{pmatrix},$$

where $(.) = -1$, $0 < \eta_i \ll 1$ and $\eta_i \notin \mathbb{Q}$. We suppose also that $\eta_i/\eta_j \notin \mathbb{Q}$, $i, j \in \{1, 2, 3\}$, $i \neq j$. An easy way to show that the semigroup $\pi< A_1, A_2, A_3 >$ is infinite is to consider the initial condition $\mathbf{e} = (e, e, e)'$ and to prove that $\Pi = \pi(< A_1, A_2, A_3 > \otimes \mathbf{e}) = \{\pi(M\mathbf{e}),\ M \in< A_1, A_2, A_3 >\}$ is infinite. We

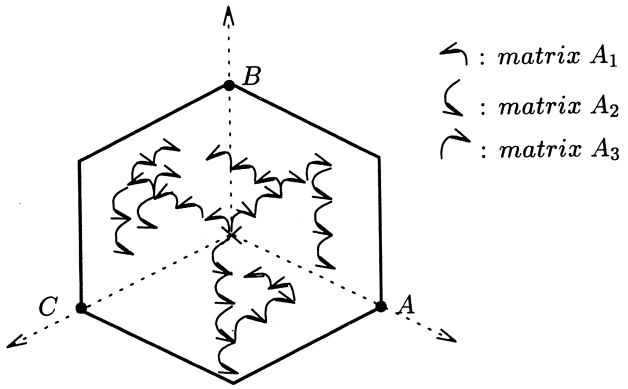


FIGURE 16. A finitely generated but projectively infinite semigroup of matrices

obtain a nice illustration of the phenomenon with the help of the graphical representation in the projective space.

The extremal eigenvectors of A_1, A_2 and A_3 are $(B,C),(A,C)$ and (A,B) respectively. The picture is analog to the one of Figure 15, with three different transient regimes interacting.

For a point $\mathbf{u} = (u_1, u_2, u_3)'$ such that $d(\mathbf{u}, \mathbf{e}) < 1 - \sup_{i=1,2,3} \eta_i$, where d is the projective distance (def. 3.4), we have:

$$A_1\mathbf{u} = \begin{pmatrix} u_1 - \eta_1 \\ u_2 \\ u_3 \end{pmatrix}, \; A_2\mathbf{u} = \begin{pmatrix} u_1 \\ u_2 - \eta_2 \\ u_3 \end{pmatrix}, \; A_3\mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \\ u_3 - \eta_3 \end{pmatrix}.$$

It is very easy to prove that Π is dense in the unit ball of the projective space (the hexagon delimited by A, B and C). In fact let us consider three integers N_1, N_2 and N_3 such that:

$$\sup_{i=1,2,3} (N_i \times \eta_i) - \inf_{i=1,2,3} (N_i \times \eta_i) < 1 .$$

Then it is quite obvious that there exists a matrix $M \in < A_1, A_2, A_3 >$, $M = A_{u_N} \otimes \cdots \otimes A_{u_1}$ with $N = N_1 + N_2 + N_3$ such that:

$$N_i = \#\{k \mid A_{u_k} = A_i\}, \; i = 1,2,3 ,$$

$$M\mathbf{e} = (-N_1 \times \eta_1, -N_2 \times \eta_2, -N_3 \times \eta_3)' .$$

In fact it is easy to understand, watching Figure 16, that we will obtain this formula for $M\mathbf{e}$ iff $\forall n \in \{1, \ldots, N\}$, $\pi(A_{u_n} \otimes \cdots \otimes A_{u_1}\mathbf{e})$ belongs to the interior of the hexagon (A, B, C).

We consider an arbitrary point $\mathbf{v}$ of the interior of the hexagon (A, B, C). As η_1, η_2, η_3 are not co-rational, there exists a sequence of integers $N^{(n)}$ and a sequence of matrices $\{M^{(n)}, \; M^{(n)} \in < A_1, \ldots, A_k >\}$ with the following properties.

- The length of $M^{(n)}$ is $N^{(n)}$, i.e $M^{(n)} = A^{(n)}_{u_{N^{(n)}}} \otimes \cdots \otimes A^{(n)}_{u_1}$.
- $N_i^{(n)} = \#\{k \mid A^{(n)}_{u_k} = A_i\}$, $i = 1, 2, 3$,

$$\pi(M^{(n)}\mathbf{e}) = \pi\left(-N_1^{(n)} \times \eta_1, -N_2^{(n)} \times \eta_2, -N_3^{(n)} \times \eta_3\right)' \stackrel{n\to\infty}{\longrightarrow} \pi(\mathbf{v}) .$$

Remark If we consider another initial condition $\mathbf{u} \neq \mathbf{e}$, we will in general obtain a set of reachable points $\pi(< A_1, \ldots, A_k > \mathbf{u})$ dense in the hexagon Π and whose intersection with Π is empty. Let us now consider a Markov chain $x(n, x_0)$ whose transition probabilities $p(.,.)$ verify:

$$\forall \mathbf{v} \in I\!R^3_{Max},\ p(\pi(\mathbf{v}), \pi(A_i\mathbf{v})) = p_i,\ i = 1, 2, 3,\ p_i > 0,\ p_1 + p_2 + p_3 = 1 .$$

We take $\mathbf{e}$ as our initial condition. Then Π is a set of transient states for the Markov chain. After the first and before the second hitting time of the border of the hexagon (A, B, C), the Markov chain evolves on a set of transient states dense in the interior of (A, B, C) and whose intersection with Π is empty. It is however possible to show that the chain is positive recurrent. Points A, B or C can be used as regenerative points (for example $\pi(A_3^k A_2^{k'} u) = A$, $\forall u \in I\!R^J$, when k and k' are sufficiently large).

5.2 Multiplicity of stationary regimes.

We consider a stochastic model of product of matrices in the $I\!R_{Max}$ algebra. The model is the following one:

$$x(n+1) \quad = \quad A(n)x(n) , \tag{5}$$

where $x(n+1)$ and $x(n)$ are $I\!R^J$-valued vectors and $A(n)$ is an irreducible random matrix of size $J \times J$. The exogenous sequence $\{A(n),\ n \in I\!N\}$ is i.i.d.

The interest for such models has been initiated by the study of Stochastic Event Graphs, a class of Stochastic Petri Networks. Many networks with synchronization and/or blocking can be modelled this way. Many examples can be found in [1] and [2].
We are interested in the stationary behaviour of $\pi x(n)$. For a network modelled by such a system, we can compute quantities such as queue length, sojourn or idle times from the knowledge of $\pi x(n)$. In [9], necessary and sufficient conditions to have a unique stationary regime for $\pi x(n)$ are given.

There is another problem worth considering. What happens for a fixed deterministic initial condition x_0, is it possible to get several stationary regimes? Another way to state the problem is the following one. For a given network with a fixed initial condition and a stochastic dynamic given by equation (5), is it possible to obtain several stationary regimes for queue length or sojourn time. The answer, rather counter-intuitive, is yes.

More precisely, for an i.i.d model $x(n+1) = A(n)x(n)$ and $x_0 \in I\!R^J$, $\pi x(n, x_0)$ is a Markov chain and this chain can have several classes of recurrence. Let us illustrate this.

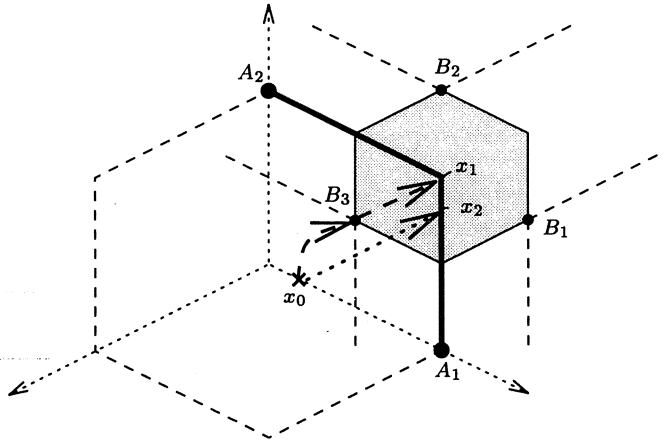


FIGURE 17. A single initial condition and several stationary regimes.

We have $A(n) = A$ or B with $P(A(0) = A) = p > 0$ and $P(A(0) = B) = 1 - p > 0$.

$$A = \begin{pmatrix} e & -2 & -2 \\ -2 & e & -2 \\ -2 & -2 & -4 \end{pmatrix},$$

$$B = \begin{pmatrix} e & -1 & 1 \\ -1 & e & 1 \\ -3 & -3 & e \end{pmatrix} = \begin{pmatrix} e & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & -2 \end{pmatrix} \otimes \begin{pmatrix} e & -1 & -1 \\ -1 & e & -1 \\ -1 & -1 & e \end{pmatrix}$$

$$\otimes \begin{pmatrix} e & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & 2 \end{pmatrix}.$$

In the previous line, we have written matrix B in a form which reflects the fact that it is exactly the matrix studied in Figure 6 up to a change of origin (see Lemma 4.2).

On Figure 17, we have materialized the domains of attraction of the matrix B. We consider the initial condition $x_0 = (\eta, e, e)'$, where $e < \eta \ll 1$. This initial condition is in the domain of attraction of B_3. As a consequence, we have $\pi(Bx_0) = B_3$ and $\pi(ABx_0) = \pi(AB^k x_0) = \pi(e, e, -2)'$. We check that $\pi(Ax_0) = \pi(e, e-\eta, -2)'$. Both vectors $x_1 = (e, e, -2)'$ and $x_2 = (e, e-\eta, -2)'$ are eigenvectors of both matrices A and B. We conclude that with probability p, we have $\pi x(n, x_0) = \pi(e, e-\eta, -2)'$ and with probability $1-p$, we have $\pi x(n, x_0) = \pi(e, e, -2)'$.
There are two absorbing states for the Markov chain $\pi x(n, x_0)$.

This counter-example would probably not have been found without the help of the graphical representation in the projective space. With Figure 17, the multiplicity of stationary regimes becomes rather clear.

6 SOFTWARE

A C program has been written by Bruno Gaujal, which implements the algorithm of Section 4.3. Given a matrix of dimension 3, this program provides the graphical representation of eigenvectors, periodic regimes and domains of attraction (as in Figures 6 to 12). If you are interested in obtaining this program, send a request to gaujal@sophia.inria.fr or mairesse@sophia.inria.fr.

Acknowledgement I would like to thank François Baccelli who introduced me to this subject and whose help has been constant since. I also thank Stéphane Gaubert for many remarks and suggestions to improve this paper.

REFERENCES

1. Baccelli F., *Ergodic theory of stochastic Petri networks*, Annals of Probability, Vol **20**, N.**1**, pp 375-396, 1992.
2. Baccelli F., Cohen G., Olsder G.J., Quadrat J.P., *Synchronization and Linearity*, John Wiley & Sons, 1993.
3. Baccelli F., Jean-Marie A., Liu Z., *A survey on solution methods for task graph models*, Proceedings of Erlangen QMIPS workshop, 1993.
4. Cohen G., Dubois D., Quadrat J.P., Viot M., *A linear system theoretic view of discrete event processes and its use for performance evaluation in manufacturing*, IEEE Trans. on Automatic Control, AC-30, p.210-220, 1985.
5. Dudnikov P., *Endomorphisms of the semimodule of bounded functions*, in Idempotent Analysis, Maslov, Samborskiĭ Editors, Advances in Soviet Mathematics, Vol **13**, AMS, 1992.
6. Gaubert S., *Théorie des systèmes linéaires dans les dioïdes*, Thèse de doctorat, Ecole Nationale Supérieure des Mines de Paris, 1992.
7. Gaubert S., *On semigroup of matrices in the (Max,+) algebra*, preprint, 1993.
8. Gondran M., Minoux M., *Valeurs propres et vecteurs propres dans les dioïdes et leur interprétation en théorie des graphes*, EDF, Bulletin de la Direction des Etudes et Recherches, Série C, Vol **2**, pp 25-41, 1977.
9. Mairesse J., *Products of irreducible random matrices in the (Max,+) algebra - Part I*, INRIA Report n^o 1939, 1993.
10. Romanovskiĭ I., *Optimization and stationary control of discrete deterministic process in dynamic programming*, Cybernetics 3, 1967.

MATHEMATICAL CENTRE TRACTS

1 T. van der Walt. *Fixed and almost fixed points.* 1963.

2 A.R. Bloemena. *Sampling from a graph.* 1964.

3 G. de Leve. *Generalized Markovian decision processes, part I: model and method.* 1964.

4 G. de Leve. *Generalized Markovian decision processes, part II: probabilistic background.* 1964.

5 G. de Leve, H.C. Tijms, P.J. Weeda. *Generalized Markovian decision processes, applications.* 1970.

6 M.A. Maurice. *Compact ordered spaces.* 1964.

7 W.R. van Zwet. *Convex transformations of random variables.* 1964.

8 J.A. Zonneveld. *Automatic numerical integration.* 1964.

9 P.C. Baayen. *Universal morphisms.* 1964.

10 E.M. de Jager. *Applications of distributions in mathematical physics.* 1964.

11 A.B. Paalman-de Miranda. *Topological semigroups.* 1964.

12 J.A.Th.M. van Berckel, H. Brandt Corstius, R.J. Mokken, A. van Wijngaarden. *Formal properties of newspaper Dutch.* 1965.

13 H.A. Lauwerier. *Asymptotic expansions.* 1966, out of print: replaced by MCT 54.

14 H.A. Lauwerier. *Calculus of variations in mathematical physics.* 1966.

15 R. Doornbos. *Slippage tests.* 1966.

16 J.W. de Bakker. *Formal definition of programming languages with an application to the definition of ALGOL 60.* 1967.

17 R.P. van de Riet. *Formula manipulation in ALGOL 60, part 1.* 1968.

18 R.P. van de Riet. *Formula manipulation in ALGOL 60, part 2.* 1968.

19 J. van der Slot. *Some properties related to compactness.* 1968.

20 P.J. van der Houwen. *Finite difference methods for solving partial differential equations.* 1968.

21 E. Wattel. *The compactness operator in set theory and topology.* 1968.

22 T.J. Dekker. *ALGOL 60 procedures in numerical algebra, part 1.* 1968.

23 T.J. Dekker, W. Hoffmann. *ALGOL 60 procedures in numerical algebra, part 2.* 1968.

24 J.W. de Bakker. *Recursive procedures.* 1971.

25 E.R. Paërl. *Representations of the Lorentz group and projective geometry.* 1969.

26 European Meeting 1968. *Selected statistical papers, part I.* 1968.

27 European Meeting 1968. *Selected statistical papers, part II.* 1968.

28 J. Oosterhoff. *Combination of one-sided statistical tests.* 1969.

29 J. Verhoeff. *Error detecting decimal codes.* 1969.

30 H. Brandt Corstius. *Exercises in computational linguistics.* 1970.

31 W. Molenaar. *Approximations to the Poisson, binomial and hypergeometric distribution functions.* 1970.

32 L. de Haan. *On regular variation and its application to the weak convergence of sample extremes.* 1970.

33 F.W. Steutel. *Preservations of infinite divisibility under mixing and related topics.* 1970.

34 I. Juhász, A. Verbeek, N.S. Kroonenberg. *Cardinal functions in topology.* 1971.

35 M.H. van Emden. *An analysis of complexity.* 1971.

36 J. Grasman. *On the birth of boundary layers.* 1971.

37 J.W. de Bakker, G.A. Blaauw, A.J.W. Duijvestijn, E.W. Dijkstra, P.J. van der Houwen, G.A.M. Kamsteeg-Kemper, F.E.J. Kruseman Aretz, W.L. van der Poel, J.P. Schaap-Kruseman, M.V. Wilkes, G. Zoutendijk. *MC-25 Informatica Symposium.* 1971.

38 W.A. Verloren van Themaat. *Automatic analysis of Dutch compound words.* 1972.

39 H. Bavinck. *Jacobi series and approximation.* 1972.

40 H.C. Tijms. *Analysis of (s,S) inventory models.* 1972.

41 A. Verbeek. *Superextensions of topological spaces.* 1972.

42 W. Vervaat. *Success epochs in Bernoulli trials (with applications in number theory).* 1972.

43 F.H. Ruymgaart. *Asymptotic theory of rank tests for independence.* 1973.

44 H. Bart. *Meromorphic operator valued functions.* 1973.

45 A.A. Balkema. *Monotone transformations and limit laws.* 1973.

46 R.P. van de Riet. *ABC ALGOL, a portable language for formula manipulation systems, part 1: the language.* 1973.

47 R.P. van de Riet. *ABC ALGOL, a portable language for formula manipulation systems, part 2: the compiler.* 1973.

48 F.E.J. Kruseman Aretz, P.J.W. ten Hagen, H.L. Oudshoorn. *An ALGOL 60 compiler in ALGOL 60, text of the MC-compiler for the EL-X8.* 1973.

49 H. Kok. *Connected orderable spaces.* 1974.

50 A. van Wijngaarden, B.J. Mailloux, J.E.L. Peck, C.H.A. Koster, M. Sintzoff, C.H. Lindsey, L.G.L.T. Meertens, R.G. Fisker (eds.). *Revised report on the algorithmic language ALGOL 68.* 1976.

51 A. Hordijk. *Dynamic programming and Markov potential theory.* 1974.

52 P.C. Baayen (ed.). *Topological structures.* 1974.

53 M.J. Faber. *Metrizability in generalized ordered spaces.* 1974.

54 H.A. Lauwerier. *Asymptotic analysis, part 1.* 1974.

55 M. Hall, Jr., J.H. van Lint (eds.). *Combinatorics, part 1: theory of designs, finite geometry and coding theory.* 1974.

56 M. Hall, Jr., J.H. van Lint (eds.). *Combinatorics, part 2: graph theory, foundations, partitions and combinatorial geometry.* 1974.

57 M. Hall, Jr., J.H. van Lint (eds.). *Combinatorics, part 3: combinatorial group theory.* 1974.

58 W. Albers. *Asymptotic expansions and the deficiency concept in statistics.* 1975.

59 J.L. Mijnheer. *Sample path properties of stable processes.* 1975.

60 F. Göbel. *Queueing models involving buffers.* 1975.

63 J.W. de Bakker (ed.). *Foundations of computer science.* 1975.

64 W.J. de Schipper. *Symmetric closed categories.* 1975.

65 J. de Vries. *Topological transformation groups, 1: a categorical approach.* 1975.

66 H.G.J. Pijls. *Logically convex algebras in spectral theory and eigenfunction expansions.* 1976.

68 P.P.N. de Groen. *Singularly perturbed differential operators of second order.* 1976.

69 J.K. Lenstra. *Sequencing by enumerative methods.* 1977.

70 W.P. de Roever, Jr. *Recursive program schemes: semantics and proof theory.* 1976.

71 J.A.E.E. van Nunen. *Contracting Markov decision processes.* 1976.

72 J.K.M. Jansen. *Simple periodic and non-periodic Lamé functions and their applications in the theory of conical waveguides.* 1977.

73 D.M.R. Leivant. *Absoluteness of intuitionistic logic.* 1979.

74 H.J.J. te Riele. *A theoretical and computational study of generalized aliquot sequences.* 1976.

75 A.E. Brouwer. *Treelike spaces and related connected topological spaces.* 1977.

76 M. Rem. *Associons and the closure statements.* 1976.

77 W.C.M. Kallenberg. *Asymptotic optimality of likelihood ratio tests in exponential families.* 1978.

78 E. de Jonge, A.C.M. van Rooij. *Introduction to Riesz spaces.* 1977.

79 M.C.A. van Zuijlen. *Empirical distributions and rank statistics.* 1977.

80 P.W. Hemker. *A numerical study of stiff two-point boundary problems.* 1977.

81 K.R. Apt, J.W. de Bakker (eds.). *Foundations of computer science II, part 1.* 1976.

82 K.R. Apt, J.W. de Bakker (eds.). *Foundations of computer science II, part 2.* 1976.

83 L.S. van Benthem Jutting. *Checking Landau's "Grundlagen" in the AUTOMATH system.* 1979.

84 H.L.L. Busard. *The translation of the elements of Euclid from the Arabic into Latin by Hermann of Carinthia (?), books vii-xii.* 1977.

85 J. van Mill. *Supercompactness and Wallmann spaces.* 1977.

86 S.G. van der Meulen, M. Veldhorst. *Torrix I, a programming system for operations on vectors and matrices over arbitrary fields and of variable size.* 1978.

88 A. Schrijver. *Matroids and linking systems.* 1977.

89 J.W. de Roever. *Complex Fourier transformation and analytic functionals with unbounded carriers.* 1978.

90 L.P.J. Groenewegen. *Characterization of optimal strategies in dynamic games.* 1981.

91 J.M. Geysel. *Transcendence in fields of positive characteristic.* 1979.

92 P.J. Weeda. *Finite generalized Markov programming.* 1979.

93 H.C. Tijms, J. Wessels (eds.). *Markov decision theory.* 1977.

94 A. Bijlsma. *Simultaneous approximations in transcendental number theory.* 1978.

95 K.M. van Hee. *Bayesian control of Markov chains.* 1978.

96 P.M.B. Vitányi. *Lindenmayer systems: structure, languages, and growth functions.* 1980.

97 A. Federgruen. *Markovian control problems; functional equations and algorithms.* 1984.

98 R. Geel. *Singular perturbations of hyperbolic type.* 1978.

99 J.K. Lenstra, A.H.G. Rinnooy Kan, P. van Emde Boas (eds.). *Interfaces between computer science and operations research.* 1978.

100 P.C. Baayen, D. van Dulst, J. Oosterhoff (eds.). *Proceedings bicentennial congress of the Wiskundig Genootschap, part 1.* 1979.

101 P.C. Baayen, D. van Dulst, J. Oosterhoff (eds.). *Proceedings bicentennial congress of the Wiskundig Genootschap, part 2.* 1979.

102 D. van Dulst. *Reflexive and superreflexive Banach spaces.* 1978.

103 K. van Harn. *Classifying infinitely divisible distributions by functional equations.* 1978.

104 J.M. van Wouwe. *GO-spaces and generalizations of metrizability.* 1979.

105 R. Helmers. *Edgeworth expansions for linear combinations of order statistics.* 1982.

106 A. Schrijver (ed.). *Packing and covering in combinatorics.* 1979.

107 C. den Heijer. *The numerical solution of nonlinear operator equations by imbedding methods.* 1979.

108 J.W. de Bakker, J. van Leeuwen (eds.). *Foundations of computer science III, part 1.* 1979.

109 J.W. de Bakker, J. van Leeuwen (eds.). *Foundations of computer science III, part 2.* 1979.

110 J.C. van Vliet. *ALGOL 68 transput, part I: historical review and discussion of the implementation model.* 1979.

111 J.C. van Vliet. *ALGOL 68 transput, part II: an implementation model.* 1979.

112 H.C.P. Berbee. *Random walks with stationary increments and renewai theory.* 1979.

113 T.A.B. Snijders. *Asymptotic optimality theory for testing problems with restricted alternatives.* 1979.

114 A.J.E.M. Janssen. *Application of the Wigner distribution to harmonic analysis of generalized stochastic processes.* 1979.

115 P.C. Baayen, J. van Mill (eds.). *Topological structures II, part 1.* 1979.

116 P.C. Baayen, J. van Mill (eds.). *Topological structures II, part 2.* 1979.

117 P.J.M. Kallenberg. *Branching processes with continuous state space.* 1979.

118 P. Groeneboom. *Large deviations and asymptotic efficiencies.* 1980.

119 F.J. Peters. *Sparse matrices and substructures, with a novel implementation of finite element algorithms.* 1980.

120 W.P.M. de Ruyter. *On the asymptotic analysis of large-scale ocean circulation.* 1980.

121 W.H. Haemers. *Eigenvalue techniques in design and graph theory.* 1980.

122 J.C.P. Bus. *Numerical solution of systems of nonlinear equations.* 1980.

123 I. Yuhász. *Cardinal functions in topology - ten years later.* 1980.

124 R.D. Gill. *Censoring and stochastic integrals.* 1980.

125 R. Eising. *2-D systems, an algebraic approach.* 1980.

126 G. van der Hoek. *Reduction methods in nonlinear programming.* 1980.

127 J.W. Klop. *Combinatory reduction systems.* 1980.

128 A.J.J. Talman. *Variable dimension fixed point algorithms and triangulations.* 1980.

129 G. van der Laan. *Simplicial fixed point algorithms.* 1980.

130 P.J.W. ten Hagen, T. Hagen, P. Klint, H. Noot, H.J. Sint, A.H. Veen. *ILP: intermediate language for pictures.* 1980.

131 R.J.R. Back. *Correctness preserving program refinements: proof theory and applications.* 1980.

132 H.M. Mulder. *The interval function of a graph.* 1980.

133 C.A.J. Klaassen. *Statistical performance of location estimators.* 1981.

134 J.C. van Vliet, H. Wupper (eds.). *Proceedings international conference on ALGOL 68.* 1981.

135 J.A.G. Groenendijk, T.M.V. Janssen, M.J.B. Stokhof (eds.). *Formal methods in the study of language, part I.* 1981.

136 J.A.G. Groenendijk, T.M.V. Janssen, M.J.B. Stokhof (eds.). *Formal methods in the study of language, part II.* 1981.

137 J. Telgen. *Redundancy and linear programs.* 1981.

138 H.A. Lauwerier. *Mathematical models of epidemics.* 1981.

139 J. van der Wal. *Stochastic dynamic programming, successive approximations and nearly optimal strategies for Markov decision processes and Markov games.* 1981.

140 J.H. van Geldrop. *A mathematical theory of pure exchange economies without the no-critical-point hypothesis.* 1981.

141 G.E. Welters. *Abel-Jacobi isogenies for certain types of Fano threefolds.* 1981.

142 H.R. Bennett, D.J. Lutzer (eds.). *Topology and order structures, part 1.* 1981.

143 J.M. Schumacher. *Dynamic feedback in finite- and infinite-dimensional linear systems.* 1981.

144 P. Eijgenraam. *The solution of initial value problems using interval arithmetic; formulation and analysis of an algorithm.* 1981.

145 A.J. Brentjes. *Multi-dimensional continued fraction algorithms.* 1981.

146 C.V.M. van der Mee. *Semigroup and factorization methods in transport theory.* 1981.

147 H.H. Tigelaar. *Identification and informative sample size.* 1982.

148 L.C.M. Kallenberg. *Linear programming and finite Markovian control problems.* 1983.

149 C.B. Huijsmans, M.A. Kaashoek, W.A.J. Luxemburg, W.K. Vietsch (eds.). *From A to Z, proceedings of a symposium in honour of A.C. Zaanen.* 1982.

150 M. Veldhorst. *An analysis of sparse matrix storage schemes.* 1982.

151 R.J.M.M. Does. *Higher order asymptotics for simple linear rank statistics.* 1982.

152 G.F. van der Hoeven. *Projections of lawless sequencies.* 1982.

153 J.P.C. Blanc. *Application of the theory of boundary value problems in the analysis of a queueing model with paired services.* 1982.

154 H.W. Lenstra, Jr., R. Tijdeman (eds.). *Computational methods in number theory, part I.* 1982.

155 H.W. Lenstra, Jr., R. Tijdeman (eds.). *Computational methods in number theory, part II.* 1982.

156 P.M.G. Apers. *Query processing and data allocation in distributed database systems.* 1983.

157 H.A.W.M. Kneppers. *The covariant classification of two-dimensional smooth commutative formal groups over an algebraically closed field of positive characteristic.* 1983.

158 J.W. de Bakker, J. van Leeuwen (eds.). *Foundations of computer science IV, distributed systems, part 1.* 1983.

159 J.W. de Bakker, J. van Leeuwen (eds.). *Foundations of computer science IV, distributed systems, part 2.* 1983.

160 A. Rezus. *Abstract AUTOMATH.* 1983.

161 G.F. Helminck. *Eisenstein series on the metaplectic group, an algebraic approach.* 1983.

162 J.J. Dik. *Tests for preference.* 1983.

163 H. Schippers. *Multiple grid methods for equations of the second kind with applications in fluid mechanics.* 1983.

164 F.A. van der Duyn Schouten. *Markov decision processes with continuous time parameter.* 1983.

165 P.C.T. van der Hoeven. *On point processes.* 1983.

166 H.B.M. Jonkers. *Abstraction, specification and implementation techniques, with an application to garbage collection.* 1983.

167 W.H.M. Zijm. *Nonnegative matrices in dynamic programming.* 1983.

168 J.H. Evertse. *Upper bounds for the numbers of solutions of diophantine equations.* 1983.

169 H.R. Bennett, D.J. Lutzer (eds.). *Topology and order structures, part 2.* 1983.

CWI TRACTS

1 D.H.J. Epema. *Surfaces with canonical hyperplane sections.* 1984.

2 J.J. Dijkstra. *Fake topological Hilbert spaces and characterizations of dimension in terms of negligibility.* 1984.

3 A.J. van der Schaft. *System theoretic descriptions of physical systems.* 1984.

4 J. Koene. *Minimal cost flow in processing networks, a primal approach.* 1984.

5 B. Hoogenboom. *Intertwining functions on compact Lie groups.* 1984.

6 A.P.W. Böhm. *Dataflow computation.* 1984.

7 A. Blokhuis. *Few-distance sets.* 1984.

8 M.H. van Hoorn. *Algorithms and approximations for queueing systems.* 1984.

9 C.P.J. Koymans. *Models of the lambda calculus.* 1984.

10 C.G. van der Laan, N.M. Temme. *Calculation of special functions: the gamma function, the exponential integrals and error-like functions.* 1984.

11 N.M. van Dijk. *Controlled Markov processes; time-discretization.* 1984.

12 W.H. Hundsdorfer. *The numerical solution of nonlinear stiff initial value problems: an analysis of one step methods.* 1985.

13 D. Grune. *On the design of ALEPH.* 1985.

14 J.G.F. Thiemann. *Analytic spaces and dynamic programming: a measure theoretic approach.* 1985.

15 F.J. van der Linden. *Euclidean rings with two infinite primes.* 1985.

16 R.J.P. Groothuizen. *Mixed elliptic-hyperbolic partial differential operators: a case-study in Fourier integral operators.* 1985.

17 H.M.M. ten Eikelder. *Symmetries for dynamical and Hamiltonian systems.* 1985.

18 A.D.M. Kester. *Some large deviation results in statistics.* 1985.

19 T.M.V. Janssen. *Foundations and applications of Montague grammar, part 1: Philosophy, framework, computer science.* 1986.

20 B.F. Schriever. *Order dependence.* 1986.

21 D.P. van der Vecht. *Inequalities for stopped Brownian motion.* 1986.

22 J.C.S.P. van der Woude. *Topological dynamix.* 1986.

23 A.F. Monna. *Methods, concepts and ideas in mathematics: aspects of an evolution.* 1986.

24 J.C.M. Baeten. *Filters and ultrafilters over definable subsets of admissible ordinals.* 1986.

25 A.W.J. Kolen. *Tree network and planar rectilinear location theory.* 1986.

26 A.H. Veen. *The misconstrued semicolon: Reconciling imperative languages and dataflow machines.* 1986.

27 A.J.M. van Engelen. *Homogeneous zero-dimensional absolute Borel sets.* 1986.

28 T.M.V. Janssen. *Foundations and applications of Montague grammar, part 2: Applications to natural language.* 1986.

29 H.L. Trentelman. *Almost invariant subspaces and high gain feedback.* 1986.

30 A.G. de Kok. *Production-inventory control models: approximations and algorithms.* 1987.

31 E.E.M. van Berkum. *Optimal paired comparison designs for factorial experiments.* 1987.

32 J.H.J. Einmahl. *Multivariate empirical processes.* 1987.

33 O.J. Vrieze. *Stochastic games with finite state and action spaces.* 1987.

34 P.H.M. Kersten. *Infinitesimal symmetries: a computational approach.* 1987.

35 M.L. Eaton. *Lectures on topics in probability inequalities.* 1987.

36 A.H.P. van der Burgh, R.M.M. Mattheij (eds.). *Proceedings of the first international conference on industrial and applied mathematics (ICIAM 87).* 1987.

37 L. Stougie. *Design and analysis of algorithms for stochastic integer programming.* 1987.

38 J.B.G. Frenk. *On Banach algebras, renewal measures and regenerative processes.* 1987.

39 H.J.M. Peters, O.J. Vrieze (eds.). *Surveys in game theory and related topics.* 1987.

40 J.L. Geluk, L. de Haan. *Regular variation, extensions and Tauberian theorems.* 1987.

41 Sape J. Mullender (ed.). *The Amoeba distributed operating system: Selected papers 1984-1987.* 1987.

42 P.R.J. Asveld, A. Nijholt (eds.). *Essays on concepts, formalisms, and tools.* 1987.

43 H.L. Bodlaender. *Distributed computing: structure and complexity.* 1987.

44 A.W. van der Vaart. *Statistical estimation in large parameter spaces.* 1988.

45 S.A. van de Geer. *Regression analysis and empirical processes.* 1988.

46 S.P. Spekreijse. *Multigrid solution of the steady Euler equations.* 1988.

47 J.B. Dijkstra. *Analysis of means in some non-standard situations.* 1988.

48 F.C. Drost. *Asymptotics for generalized chi-square goodness-of-fit tests.* 1988.

49 F.W. Wubs. *Numerical solution of the shallow-water equations.* 1988.

50 F. de Kerf. *Asymptotic analysis of a class of perturbed Korteweg-de Vries initial value problems.* 1988.

51 P.J.M. van Laarhoven. *Theoretical and computational aspects of simulated annealing.* 1988.

52 P.M. van Loon. *Continuous decoupling transformations for linear boundary value problems.* 1988.

53 K.C.P. Machielsen. *Numerical solution of optimal control problems with state constraints by sequential quadratic programming in function space.* 1988.

54 L.C.R.J. Willenborg. *Computational aspects of survey data processing.* 1988.

55 G.J. van der Steen. *A program generator for recognition, parsing and transduction with syntactic patterns.* 1988.

56 J.C. Ebergen. *Translating programs into delay-insensitive circuits.* 1989.

57 S.M. Verduyn Lunel. *Exponential type calculus for linear delay equations.* 1989.

58 M.C.M. de Gunst. *A random model for plant cell population growth.* 1989.

59 D. van Dulst. *Characterizations of Banach spaces not containing l^1.* 1989.

60 H.E. de Swart. *Vacillation and predictability properties of low-order atmospheric spectral models.* 1989.

61 P. de Jong. *Central limit theorems for generalized multilinear forms.* 1989.

62 V.J. de Jong. *A specification system for statistical software.* 1989.

63 B. Hanzon. *Identifiability, recursive identification and spaces of linear dynamical systems, part I.* 1989.

64 B. Hanzon. *Identifiability, recursive identification and spaces of linear dynamical systems, part II.* 1989.

65 B.M.M. de Weger. *Algorithms for diophantine equations.* 1989.

66 A. Jung. *Cartesian closed categories of domains.* 1989.

67 J.W. Polderman. *Adaptive control & identification: Conflict or conflux?.* 1989.

68 H.J. Woerdeman. *Matrix and operator extensions.* 1989.

69 B.G. Hansen. *Monotonicity properties of infinitely divisible distributions.* 1989.

70 J.K. Lenstra, H.C. Tijms, A. Volgenant (eds.). *Twenty-five years of operations research in the Netherlands: Papers dedicated to Gijs de Leve.* 1990.

71 P.J.C. Spreij. *Counting process systems. Identification and stochastic realization.* 1990.

72 J.F. Kaashoek. *Modeling one dimensional pattern formation by anti-diffusion.* 1990.

73 A.M.H. Gerards. *Graphs and polyhedra. Binary spaces and cutting planes.* 1990.

74 B. Koren. *Multigrid and defect correction for the steady Navier-Stokes equations. Application to aerodynamics.* 1991.

75 M.W.P. Savelsbergh. *Computer aided routing.* 1992.

76 O.E. Flippo. *Stability, duality and decomposition in general mathematical programming.* 1991.

77 A.J. van Es. *Aspects of nonparametric density estimation.* 1991.

78 G.A.P. Kindervater. *Exercises in parallel combinatorial computing.* 1992.

79 J.J. Lodder. *Towards a symmetrical theory of generalized functions.* 1991.

80 S.A. Smulders. *Control of freeway traffic flow.* 1993.

81 P.H.M. America, J.J.M.M. Rutten. *A parallel object-oriented language: design and semantic foundations.* 1992.

82 F. Thuijsman. *Optimality and equilibria in stochastic games.* 1992.

83 R.J. Kooman. *Convergence properties of recurrence sequences.* 1992.

84 A.M. Cohen (ed.). *Computational aspects of Lie group representations and related topics. Proceedings of the 1990 Computational Algebra Seminar at CWI, Amsterdam.* 1991.

85 V. de Valk. *One-dependent processes.* 1994.

86 J.A. Baars, J.A.M. de Groot. *On topological and linear equivalence of certain function spaces.* 1992.

87 A.F. Monna. *The way of mathematics and mathematicians.* 1992.

88 E.D. de Goede. *Numerical methods for the three-dimensional shallow water equations.* 1993.

89 M. Zwaan. *Moment problems in Hilbert space with applications to magnetic resonance imaging.* 1993.

90 C. Vuik. *The solution of a one-dimensional Stefan problem.* 1993.

91 E.R. Verheul. *Multimedians in metric and normed spaces.* 1993.

92 J.L.M. Maubach. *Iterative methods for non-linear partial differential equations.* 1993.

93 A.W. Ambergen. *Statistical uncertainties in posterior probabilities.* 1993.

94 P.A. Zegeling. *Moving-grid methods for time-dependent partial differential equations.* 1993.

95 M.J.C. van Pul. *Statistical analysis of software reliability models.* 1993.

96 J.K. Scholma. *A Lie algebraic study of some integrable systems associated with root systems.* 1993.

97 J.L. van den Berg. *Sojourn times in feedback and processor sharing queues.* 1993.

98 A.J. Koning. *Stochastic integrals and goodness-of-fit tests.* 1993.

99 B.P. Sommeijer. *Parallelism in the numerical integration of initial value problems.* 1993.

100 J. Molenaar. *Multigrid methods for semiconductor device simulation.* 1993.

101 H.J.C. Huijberts. *Dynamic feedback in nonlinear synthesis problems.* 1994.

102 J.A.M. van der Weide. *Stochastic processes and point processes of excursions.* 1994.

103 P.W. Hemker, P. Wesseling (eds.). *Contributions to multigrid.* 1994.

104 I.J.B.F. Adan. *A compensation approach for queueing problems.* 1994.

105 O.J. Boxma, G.M. Koole (eds.). *Performance evaluation of parallel and distributed systems - solution methods. Part 1.* 1994.

106 O.J. Boxma, G.M. Koole (eds.). *Performance evaluation of parallel and distributed systems - solution methods. Part 2.* 1994.