Backward Coupling in Bounded Free-Choice Nets under Markovian and non-Markovian Assumptions

Anne Bouillard *and Bruno Gaujal †

November 23, 2007

Abstract

In this paper, we show how to design a perfect sampling algorithm for stochastic Free-Choice Petri nets by backward coupling. For Markovian event graphs, the simulation time can be greatly reduced by using extremal initial states, namely blocking marking, although such nets do not exhibit any natural monotonicity property. Another approach for perfect simulation of non-Markovian event graphs is based on a (max,plus) representation of the system and the theory of (max,plus) stochastic systems. We also show how to extend this approach to one-bounded free choice nets to the expense of keeping all states. Finally, experimental runs show that the (max,plus) approach needs a larger simulation time than the Markovian approach.

1 Introduction

Petri nets can be used as alternatives to queueing systems with fork and join nodes to model communication networks involving some synchronization schemes such as networks with window control, Kanban systems or finite queues with general blocking [Baccelli et al., 1995]. Under Markovian assumptions, it can be shown that such networks are multidimensional Continuous Time Markov Chains. In the presence of fork and join nodes the steady state distribution is not a product form in general and the only general technique to compute the stationary distribution

*Anne Bouillard is with IRISA/ENS Cachan (Bretagne), Campus de Ker Lann, 35170 BRUZ, France. Email: Anne.Bouillard@bretagne.ens-cachan.fr.

†INRIA and Lab. LIG (CNRS, INPG, UJF), 51, Av. J. Kunztmann, Montbonnot, France. Email: Bruno.Gaujal@imag.fr.

is to solve the Kolmogorov equations. When the number of nodes on the net grows, the state space explodes exponentially and the computation of the stationary distribution cannot be done numerically.

Simulation approaches are alternative methods to estimate the stationary behavior of such systems by providing samples distributed according to the stationary distribution, even when it is impossible to compute this distribution numerically. Propp and Wilson used backward coupling ([Propp and Wilson, 1996]) to derive an algorithm to get perfect sampling (*i.e.* which distribution is exactly stationary) of the state of discrete time finite Markov chains. In this paper, we adapt their algorithm for *Markovian free choice Petri nets*. When the network is an *event graph*, we show how to improve drastically simulation time by reducing the number of initial states to be simulated. Event graphs do not have classical monotonicity properties: the state space does not contain any natural minimal state (*e.g.* all buffers are empty) nor a maximal state (*e.g.* all buffers are full), unlike some open networks with blocking and rejection (see [Vincent and Marchand, 2004]). However, it is possible to exhibit extremal initial states (called *blocking markings* later) such that whenever coupling from the past occurs starting from those states, the coupling state is distributed according to the stationary distribution of the net. When the network has Q transitions, these extremal states are obtained by blocking the firings of one transition and let the system evolve until a deadlock is reached. Doing this, one gets one of the Q blocking states of the network.

A second method for perfect sampling, based on a (max,plus) representation of the dynamics of the event graph, is also given. This method works under more general stochastic assumptions (basically under i.i.d. assumptions with general distributions) and does not need the network to be Markovian. In this general case, numerical computation of the stationary distribution is in general impossible even when the net has a small number of nodes, and getting stationary samples is even more critical.

The perfect sampling algorithm presented here uses the theory of (max,plus) stochastic systems developed in [Baccelli et al., 1992, Mairesse, 1997, Baccelli and Mairesse, 1998]. This theory has been used mainly to prove *existence* theorems in full generality [Baccelli and Mairesse, 1998]. To the best of our knowledge, this is the first time it is applied to design perfect simulation algorithms.

We compare the two methods for perfect simulation of event graphs. It is interesting to notice that while the (max,plus) algorithm couples faster than the Markov chain algorithm, the simulation lasts longer with the (max,plus) method because each step involves large matrix products.

The (max,plus) method is also generalized to the simulation of stochastic *heaps of pieces* with general distributions (keeping the same low complexity as for event graphs) and finally to *safe free choice nets*. However, for safe free choice nets, we lose the feature that only a small number of trajectories have to be simulated. It is interesting to notice that this is the same gap as between Markovian event graphs and Markovian free choice nets, studied in the first part of the paper. This paper is an extended version of [Bouillard and Gaujal, 2006], presented at the Valuetools conference.

2 Perfect sampling of Finite Markov Chains

In this section, we recall the main ingredients for perfect sampling of finite Markov chains useful for the following.

Let $\{X_n\}_{n\in\mathbb{N}}$ be an irreducible and aperiodic discrete time Markov chain with a finite state space $\mathcal{S}$ and a transition matrix $P = (p_{i,j})$. The evolution of the Markov chain can always be described by a stochastic recurrence sequence

$$X_{n+1} = \phi\left(X_n, u_{n+1}\right), \tag{1}$$

with X_n the state of the chain at time n and $\{u_n\}_{n\in\mathbb{Z}}$ an independent and identically distributed sequence of real random variables, uniformly distributed over $[0, 1]$. The transition function $\phi : \mathcal{S}\mathcal{E} \to \mathcal{S}$ verifies the property that $\mathbb{P}\left(\phi(i,u) = j\right) = p_{i,j}$ for every pair of states $(i, j) \in \mathcal{S}$ and for any u, a real random variable, uniformly distributed over $[0, 1]$.

Let $\phi^n : \mathcal{S}\mathcal{E}^n \to \mathcal{S}$ denote the function whose output is the state of the chain after n iterations and starting in state $s \in S$. That is, $\phi^n\left(s, u_{1\to n}\right) = \phi\left(\ldots\phi\left(\phi\left(s, u_1\right), u_2\right), \ldots, u_n\right)$. This notation can be extended to set of states. So for a set of states $A \subset \mathcal{S}$, we denote $\phi^n\left(A, u_{1\to n}\right) = \{\phi^n\left(s, u_{1\to n}\right), s \in A\}$. In the following, $|X|$ denotes the size of set X.

Theorem 1 ([Propp and Wilson, 1996])**.** *Let ϕ be a transition function on $\mathcal{S}\mathcal{E}$. There exists an integer ℓ^* such that*

$$\lim_{n\to\infty} \left|\phi^n\left(\mathcal{S}, u_{-n+1\to 0}\right)\right| = \ell^* \text{ almost surely.}$$

The system *couples* if $\ell^* = 1$. The coupling property is closely related to renovation properties of Markov chains [Foss et al., 1998]. In particular, the main result of the backward scheme is the following theorem, justifying the name "Perfect Sampling".

Theorem 2 ([Propp and Wilson, 1996])**.** *Provided that the system couples, the state when coupling occurs for the backward scheme, is steady state distributed.*

¿From this fact, a general Perfect Sampling Algorithm (PSA), sampling the steady state can be constructed. The perfect simulation algorithm that we will be using in Sections 2, 3 and 4 is given in Algorithm 1.

Algorithm 1: Perfect Simulation Algorithm (PSA) of Markov chains

---

Data: A recurrent representation ϕ of an ergodic finite Markov chain:

$X_{n+1} = \phi(X_n, U_{n+1})$, a sequence of increasing integers $N_1, N_2 \ldots$ and a sequence $U_0, U_{-1}, U_{-2}, \ldots$ of i.i.d. r.v. uniformly distributed over $[0, 1]$.

Result: At state $s \in \mathcal{S}$ generated according to the limit distribution of the Markov chain represented by ϕ.

begin
    $m := 1$;
    repeat
        forall *state* $s \in \mathcal{S}$ **do**
            Compute $X_{n+1} = \phi(X_n, U_{n+1})$, starting at time $-N_m$ with initial state s, up to time 0 using the random variables $U_{-N_m+1}, \cdots, U_0$.
        $m := m + 1$;
    until *all simulations end up in the same state (X)* ;
    return X
end

Note that the same variables $U_0, U_{-1}, U_{-2}, \cdots$ are used for all the simulations. The infinite sequence of increasing integers $(N_m)_{m \in \mathbb{N}}$ corresponds to the lengths of the simulations at each step m. The output of the algorithm does not depend on the choice of the N_m, however, the duration of the algorithm does. The optimal choice of the sequence N_m is discussed in Section 6. Also note that a given Markov transition matrix has many constructions under the form of a recurrence equation $x_{n+1} = \phi(x_n, u_{n+1})$. Using Borel-Cantelli arguments, it is possible to show [Häggström, 2002] that for each construction, the perfect simulation algorithm will terminate with a fixed probability T, where T is 1 (or 0). Therefore, for a given representation, ϕ, it is usually possible to show that the algorithm terminates for each run (or never stops). This makes it well suited (or not) for PSA.

3 Sampling in Markovian Free-Choice Petri Nets

A Stochastic *Petri net* is a tuple $\mathcal{N} = (\mathcal{P}, \mathcal{Q}, \mathcal{F}, M_0, \tau)$ where $(\mathcal{P}, \mathcal{Q}, \mathcal{F})$ is a directed bipartite graph with nodes $\mathcal{P} \cup \mathcal{Q}$, $\mathcal{P} \cap \mathcal{Q} = \emptyset$, and arcs $\mathcal{F} \subset (\mathcal{P}\mathcal{Q}) \cup (\mathcal{Q}\mathcal{P})$ and where $M_0 \in \mathbb{N}^{\mathcal{P}}$. The elements of $\mathcal{P}$ are called *places* and those of $\mathcal{Q}$, *transitions*, and M_0 is called the *initial marking* of $\mathcal{N}$. The set of all reachable *states* (or *markings*) from M_0 is $\mathcal{R}(M_0)$. The marking $M(p)$ of a place p is also called the number of *tokens* present in place p. For a node $x \in \mathcal{P} \cup \mathcal{Q}$, we denote by ${}^\bullet x$ the set of its predecessors (inputs) and by $x^\bullet$ the set of its successors (outputs). The net is *Free-Choice* when for all $q_1, q_2 \in \mathcal{Q}$, ${}^\bullet q_1 \cap {}^\bullet q_2 \in \{\emptyset, {}^\bullet q_1\}$. The net is an *Event Graph* if all places have a single input and a single output transition: $\forall p \in \mathcal{P}$, $|{}^\bullet p| = |p^\bullet| = 1$. Note that event graphs are a strict subset of free-choice nets.

Markings evolve in time by *firing* transitions. Starting from marking M_1, a transition q is *enabled* if the marking of each input place is not null: $\forall p \in {}^\bullet q, M_1(p) > 0$. The firing of an enabled transition q removes a token in each input place (in ${}^\bullet q$) and adds a token in each output place (in $q^\bullet$) to obtain the new marking M_2. This is denoted by $M_1 \xrightarrow{q} M_2$. Since markings cannot be negative, firing q is forbiden if one input place has a null marking.

The *firing times* attached to all the transitions are the durations of the firings. They form sequences $\tau = (\tau_q)_{q \in \mathcal{Q}}$ of random sequences, $\tau_q = (\tau_q(n))_{n \in \mathbb{Z}}$ of i.i.d. variables with finite expectations ($E(\tau_q(1)) < \infty$).

In the following, we consider a Free-Choice Petri net with P places, Q transitions and Γ clusters (*i.e.* minimal sets of places G and transitions H such that $H = G^\bullet$) which is *bounded* (the total number of tokens present in the system cannot exceed some bound B) and *live* (no deadlock or starvation can ever occur). In particular, the closeness of the network is implied by the boundness and live assumptions (see [Desel and Esparza, 1995]). We assume that all transitions q have *self loops*: ${}^\bullet q \cap q^\bullet$ is a singleton $\{p\}$ with $M_0(p) = 1$.This gives a single-server semantics to all transitions[1]. We also assume that firings occur using the *race policy* (*i.e.* if two or more transitions in the same cluster are enabled, the one with the smallest firing time fires, and removes the tokens in their common input places while the firings of all the other transitions fail). For more details and precise definitions on Petri nets, we refer to [Desel and Esparza, 1995].

In this section , we consider *exponentially distributed firing times* for transitions with parameters $\lambda_q, q \in \mathcal{Q}$. In this case, the evolution of the marking M of the system can be written under the

[1] Throughout the paper, the self loops are not displayed in the figures in the sake of clarity. The only Petri net without implicit self-loops is displayed in Figure 3

form of a finite continuous time Markov chain over the set $cR(M_0)$ which infinitesimal generator is $W = (W_{M_1,M_2 \in \mathcal{R}\mathcal{R}})$ with $W_{M_1,M_2} =$

$$\begin{cases} \lambda_q & \text{if } M_1 \xrightarrow{q} M_2 \\ 0 & \text{otherwise} \\ -\sum_{M' \neq M_1} W_{M_1,M'} & \text{if } M_1 = M_2. \end{cases}$$

To construct a perfect simulation, this continuous time Markov chain can be uniformized. The usual uniformization coefficient $\sup_M\{\sum_{M' \neq M} W_{M,M'}\}$ does not provide a discrete time Markov chain easily amenable to perfect simulation. The trick here is to choose $\Lambda = \sum_q \lambda_q$ (the total event rate) instead. Although this may result into a loss of efficiency for the uniformization (in general $\Lambda > \sup_M\{\sum_{M' \neq M} W_{M,M'}\}$), this choice makes it possible to find a recurrence equation that defines a discrete time Markov chain with the same stationary distribution as the initial continuous time chain and for which PSA terminates in finite time with probability one.

Let us consider a Markov chain Z_n defined over $\mathcal{R}(M_0)$ such that

$$Z_{n+1} = \phi(Z_n, u_{n+1}), \tag{2}$$

with $(u_n)_{n \in \mathbb{N}}$ i.i.d. uniformly distributed over $[0,1]$ and ϕ defined as follows. After numbering all transitions,

$$\text{if } u \in \left[\frac{\sum_{j=1}^{i-1} \lambda_j}{\Lambda}, \frac{\sum_{j=1}^{i} \lambda_j}{\Lambda}\right), \text{ then } \quad \phi(M,u) = \begin{cases} M' & \text{if } M \xrightarrow{q_i} M' \\ M & \text{if } q_i \text{ not enabled in } M. \end{cases}$$

Using this definition of ϕ, each transition q_i is associated with an interval $I_i = \left[\frac{\sum_{j=1}^{i-1} \lambda_j}{\Lambda}, \frac{\sum_{j=1}^{i} \lambda_j}{\Lambda}\right)$. At step k, a firing of q_i occurs if $u_k \in I_i$ and if q_i is enabled under state M_{k-1}.

Theorem 1. *The Perfect Simulation Algorithm in Figure 1 based on recurrence equation (2) terminates in finite time, with probability one.*

Proof. First, it should be clear that the Markov chain Z_n has a finite state space by boundedness. This chain is aperiodic because $\phi(M,u) = M$ with positive probability (if u falls in an interval corresponding to the firing of a transition which is not enabled under M) and is irreducible because the network does not contain any deadlock (all markings are reachable from each other through some sequence of firings). Hence Z_n is ergodic.

The rest of the proof is based on the following property of the chain Z_n. For any couple of states M_1, M_2 in $\mathcal{R}(M_0)$, there exists a finite random variable k such that the chain starting in M_1

and the chain starting in M_2 reach the same state after k steps, with positive probability. Since the state space is finite, this means that starting with all possible states, the simulation reaches a unique state after a finite number of steps with positive probability (by coupling the states one by one). The result then follows using Borel-Cantelli arguments (see [Vincent and Marchand, 2004] for more on this).

To prove convergence after k steps of two chains, starting with M_1 and M_2, one can use the notion of *blocking states*. The blocking state B_a for transition a is defined as the marking in $\mathbb{R}(M_0)$ where the set of transitions that can be fired under B_a is reduced to the single transition a.

It has been proved in [Gaujal et al., 2003] that for the class of bounded and live Free-Choice Petri nets, such states exists, are unique and that B_a is reachable from any marking in $\mathcal{R}$ without ever firing transition a.

For more on blocking states (in particular on their regeneration properties, see [Gaujal et al., 2003]).

Now, we are ready for the end of the proof. Pick a arbitrarily, and consider the associated blocking state B_a. There exists a sequence of firing events that leads from M_1 to B_a. Let us consider the corresponding sequence of intervals $I_1, \dots I_\ell$. If $u_1 \in I_1, \dots, u_\ell \in I_\ell$ then $\phi^\ell(M_1, u_1, \dots, u_\ell) = B_a$. Under the same random sequence, but starting from M_2, we get $M_3 = \phi^\ell(M_2, u_1 \dots u_\ell)$. Now, starting from M_3, there exists a sequence of firings (not including a) that leads to B_a. The corresponding sequence of intervals $I_{\ell+1} \dots I_k$ are such that if $u_{\ell+1} \in I_{\ell+1}, \dots, u_k \in I_k$, $\phi^k(M_2, u_1 \dots u_k) = \phi^{k-\ell}(M_3, u_{\ell+1} \dots u_k) = B_a$ and $\phi^k(M_1, u_1 \dots u_k) = \phi^{k-\ell}(B_a, u_{\ell+1} \dots u_k) = B_a$, since under the sequence $u_{\ell+1}, \dots, u_k$, transition a never fires so that no state change can happen starting in state B_a.

Such a sequence $u_1, \dots, u_k$ occurs with positive probability ($\prod_{i=1}^k |I_i| > 0$). This finishes the proof.

$\square$

The problem with this perfect simulation scheme is that one needs to start with all states in $\mathcal{R}$ and look for coupling at time 0. the initial marking is the blocking marking of q_i, for all $i > 1$. The size of $\mathcal{R}$ can be exponential in the size of the net so that only small nets can be sampled using this approach. In the following, we will show how to reduce the number of starting states. This will only work for event graphs.

4 Sampling of Event Graphs

Here, we consider a Markovian *event graph* which is bounded and has no deadlock. With no loss of generality we may assume that between two transitions q and s, there is at most one place when the net is an event graph. If such a place exists, we denote by $M(q, s)$ the number of tokens in that place under state M.

We will show in the following that starting the simulation with the blocking states only, $\{B_a, a \in \mathcal{Q}\}$, will provide a perfect sampling when coupling occurs.

4.1 Blocking Markings

Theorem 2. *Consider a bounded event graph with no deadlock, with exponential firing times. The perfect simulation algorithm given in 1 using only blocking markings as starting points terminates in finite time with probability one and outputs a state distributed according to the stationary distribution of Z_n.*

Before we prove this theorem, which is the main result of this section, let us first make several comments.

First, this result means that one can run the perfect simulation algorithm starting with the blocking states only. This decreases the number of sample paths from an exponential number to a linear number (there is one blocking state per transition).

Another remark is that, although event graphs do not exhibit usual monotonicity properties (such as open networks with finite queues, see [Vincent and Marchand, 2004]), they do possess extremal states in some sense: the blocking markings.

The proof of the theorem comes in several steps. Let us first state a structural lemma. If σ is a sequence of firings, s a transition and M a state of the net, we denote by $N_s(\sigma, M)$ the number of times s fires when starting from M and trying to fire the sequence of firings σ. More precisely, denote M_{k-1} the marking reached just before considering the firing of σ_k. If σ_k is enabled, then it is fired: $M_{k-1} \xrightarrow{\sigma_k} M_k$; otherwise nothing happens $M_k = M_{k-1}$. This is also called *firing σ from M*. The number of times transition s is fired in this process is $N_s(\sigma, M)$.

Let M be a state under which only two transitions are enabled, say a and b. Let us denote by σ_a a shortest sequence of firings not including a, that leads from M to B_a, and by σ_b a shortest firing sequence not including b that leads from M to B_b. One knows that two such sequences exist (but are not necessarily unique) according to [Gaujal et al., 2003].

Lemma 3. *Under the foregoing notations, if σ is an arbitrary sequence of firings and s an arbitrary transition,*

$$N_s(\sigma, M) \quad = \quad (N_s(\sigma_a, M) + N_s(\sigma, B_a)) \wedge (N_s(\sigma_b, M) + N_s(\sigma, B_b)).$$

Proof. The proof goes by induction on the length of σ. If $|\sigma| = 0$, then it is enough to show that the supports of σ_a and σ_b are disjoint. First, since only a and b are allowed under M, the first firing in σ_a must be b because σ_a does not contain a and the first firing in σ_b must be a for a similar reason. Let q be the first common transition firing in both σ_a and in σ_b. By assumption, $q \notin \{a, b\}$. Since q was not allowed under M, then some other firings must have brought tokens in the incoming places of q. But these firings must have occurred in both σ_a and σ_b, contradicting the fact that q was the first common firing. This ends the case $|\sigma| = 0$.

Now, we assume that the lemma holds for all sequences of length n, and we consider a sequence σ such that $|\sigma| = n+1$. If $s \neq q$, $N_s(\sigma' q, M) = N_s(\sigma', M)$, $N_s(\sigma, B_a) = N_s(\sigma', B_a)$ and $N_s(\sigma, B_b) = N_s(\sigma, B_a)$ and the equlity holds for σ. Therfore, The only case worth considering is when $\sigma = \sigma' s$.

Remark that for a place between q and s, an initial marking M and a firing sequence σ, $N_q(\sigma, M) + M(q, s)$ is the number of tokens that have ever been in that place and $N_s(\sigma, M)$ is the number of tokens that have been consumed. If

$$\min_{q \in {}^{\bullet\bullet}s} (N_q(\sigma', M) + M(q, s)) > N_s(\sigma', M), \tag{3}$$

then every input place of s is marked after the firing of σ' and $N_s(\sigma, M) = N_s(\sigma', M) + 1$. Otherwise, $\min_{q \in {}^{\bullet\bullet}s}(N_q(\sigma', M) + M(q, s)) = N_s(\sigma', M)$ necessarily holds and there exists an input place of s that contains no token: $N_s(\sigma, M) = N_s(\sigma', M)$.

As for B_a, $M \stackrel{\sigma_a}{\to} B_a$ so that $B_a(q, s) = M(q, s) + N_q(\sigma_a, M) - N_s(\sigma_a, M)$. Using this equality and the induction hypothesis,

$$\begin{aligned}
\min_{q \in {}^{\bullet\bullet}s} (N_q(\sigma', M) + M(q, s)) \quad &= \quad \min_{q \in {}^{\bullet\bullet}s} \big[N_q(\sigma', B_a) + N_q(\sigma_a, M) + M(q, s) \\
&\qquad \wedge \quad N_q(\sigma', B_b) + N_q(\sigma_b, M) + M(q, s)\big] \\
&= \quad \min_{q \in {}^{\bullet\bullet}s} \big[N_q(\sigma', B_a) + N_s(\sigma_a, M) + B_a(q, s) \\
&\qquad \wedge \quad N_q(\sigma', B_b) + N_s(\sigma_b, M) + B_b(q, s)\big] \\
&= \quad \big[N_s(\sigma_a, M) + \min_{q \in {}^{\bullet\bullet}s} (N_q(\sigma', B_a) + B_a(q, s))\big] \\
&\qquad \wedge \quad \big[N_s(\sigma_b, M) + \min_{q \in {}^{\bullet\bullet}s} (N_q(\sigma', B_b) + B_b(q, s))\big].
\end{aligned} \tag{4}$$

Now, if Equation (3) is satisfied and if $N_s(\sigma', M) = N_s(\sigma_a, M) + N_s(\sigma', B_a)$, then $N_s(\sigma_a, M) + \min_q(N_q(\sigma', B_a) + B_a(q, s)) > N_s(\sigma_a, M) + N_s(\sigma', B_a)$ implying $\min_q(N_q(\sigma', B_a) + B_a(q, s)) >$

$N_s(\sigma', B_a)$. Then $N_s(\sigma, B_a) = N_s(\sigma', B_a) + 1$. Then the same holds for b if $N_s(\sigma', M) = N_s(\sigma_b, M) + N_s(\sigma, B_b)$. This ends the proof under condition (3).

The only remaining case is when condition (3) is not satisfied. First, without loss of generality (by symmetry over a and b) one may assume that in (4),

$$\min_q (N_q(\sigma', M) + M(q, s)) = N_s(\sigma_a, M) + \min_q (N_q(\sigma', B_a) + B_a(q, s)).$$

If Equation (3) is not satisfied, then

$$\min_q (N_q(\sigma', M) + M(q, s)) = N_s(\sigma', M) \leq N_s(\sigma_a, M) + N_s(\sigma', B_a).$$

Combining those two equations yields $N_s(\sigma', B_a) \geq \min_q(N_q(\sigma', B_a) + B_a(q, s))$, meaning that this is actually an equality and that a is not fired.

Therefore $N_s(\sigma, M)$ increases by one if and only if the minimum of $N_s(\sigma, B_a)$ and $N_s(\sigma, B_b)$ increases by one. This ends the proof.

□

Now, we generalize to the general case: under state M an arbitrary number of transitions are allowed. We partition the set of all allowed transitions into two disjoint sets S_1 and S_2. Starting from M, let σ_1 (resp. σ_2) the shortest firing sequence (containing no transition in S_1 (resp. S_2) that leads to a blocking state B_1 (resp. B_2) for S_1 (resp. S_2), where only firings in S_1 (resp. S_2) are allowed.

The same method used in the proof of Lemma 3 can be used to show the following result.

Lemma 4. *If σ is an arbitrary sequence of firings and s an arbitrary transition,*

$$N_s(\sigma, M) \quad = \quad (N_s(\sigma_1, M) + N_s(\sigma, B_1)) \wedge (N_s(\sigma_2, M) + N_s(\sigma, B_2)). \tag{5}$$

We are now ready for the proof of the theorem.

Proof. (of Theorem 2) First, it should be obvious that the simulation starting with blocking states terminates with probability one, since they form a subset of all states. Let us assume that coupling from the past for all blocking states occurs after k steps. The corresponding sequence of firings is denoted σ and the coupling state is denoted by C.

Let M be any initial state. The proof that the simulation starting from M also couples in k steps holds by induction on the number m of firings allowed under M.

If $m = 1$ then M is a blocking state and the result holds by definition of k.

If $m > 1$ we split the set S of transitions allowed under M into $S_1 = \{a\}$ and $S_2 = S \setminus \{a\}$. As in Lemma 4, we consider the states B_1 and B_2. Using the induction assumption, the simulation starting in B_2 ends up in C after firing the sequence σ from B_2. Using the fact that B_1 is the blocking state of transition a, the simulation starting in B_1 has also reached C after firing the sequence σ. Therefore, $N_s(\sigma_1, M) + N_s(\sigma u, B_1) = N_s(\sigma_2 u, M) + N_s(\sigma, B_2)$ for all sequence u and all transition s. Using, Lemma 4, this means that for all sequence u and all transition s, $N_s(\sigma u, M) = N_s(\sigma_1, M) + N_s(\sigma u, B_1)$ In other words, after σ has been fired, a new firing in u occurs when starting in M or in B_1 together or for none of them. This is true for any sequence of firings. This means that the state reach after running σ is the same, that is equals C. □

One interesting corollary of this result is the fact that one can estimate all stationary functionals of interest with a good confidence interval using the central limit theorem by merely running several independent simulations.

4.2 Minimality of blocking markings

In the previous section, we exhibited a set of extremal markings. For some nets, this set is minimal in the sense that there is no proper subset of the blocking markings that are enough to perform the perfect sampling algorithm, but this is not always the case.

On the one hand, one can find examples for which that set is minimal. The net showed in Figure 1 is such an example. This net has k transitions, and 1 token. After firing the sequence $q_2, q_3, \ldots, q_{k-1}$ from the blocking markings of all q_i, $i \neq 1$, the marking reached is the blocking marking B_k of transition q_k. As q_1 is not fired, the simulation starting with the blocking marking B_1 of transition q_1 stays at state B_1. Therefore, starting a simulation in the blocking marking B_1 is necessary to make sure all states have coupled. By symmetry, the blocking marking of every transition is also necessary.

On the other hand, the net in Figure 2 is an example for which the set of the blocking marking is not minimal. Indeed, the blocking marking B_1 of transition q_1 $((a, c))$ and B_3 of q_3 $((b, d))$ are enough and the blocking marking of transition q_2 $((b, c))$ is useless. This can be shown by considering all possible coupling sequences starting from all initial states and noticing that all states couple if and only if B_1 and B_3 do.

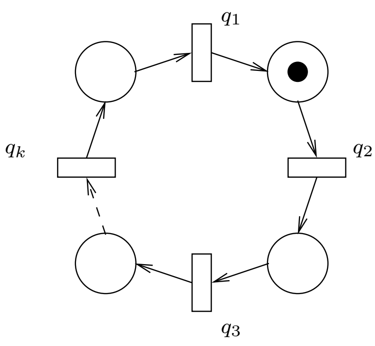

Figure 1: The set of the blocking markings can be minimal.

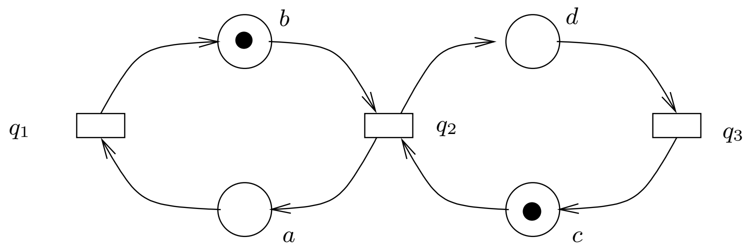

Figure 2: The set of the blocking markings is not always minimal.

4.3 Counterexamples for more general cases

In this section, we show that blocking states are no longer extremal states for the simulation in more general cases. They may couple into a state which is not distributed according to the stationary distribution.

4.3.1 Event graphs without self-loops

The general simulation scheme can be readily adapted for transitions with no self-loops. When a transition does not have a self loop, it may fire several times simultaneously under marking M (using several tokens in each input place). This number is denoted $e_q(M)$. The *enabling degree* e_q of a transition q is the maximal number of simultaneous firing that q can initiate over all markings in $\mathcal{R}(M_0)$: $e_q = \max_{M \in \mathcal{R}(M_0)} e_q(M)$. It is bounded because $\mathcal{R}(M_0)$ is finite.

The marking evolution can still be described using a continuous time Markov chain Z_n defined over $\mathcal{R}(M_0)$ using the transition function. $Z_{n+1} = \phi(Z_n, u_{n+1})$ with $(u_n)_{n\in\mathbb{N}}$ i.i.d. uniformly distributed over $[0, 1]$ and ϕ is defined as follows. After numbering all transitions, let $\Lambda = \sum_{j=1}^{q} e_j \lambda_j$.

If $0 < d \leq e_i$ and if $u \in \left[\frac{\sum_{j=1}^{i-1} e_j\lambda_j+(d-1)\lambda_i}{\Lambda}, \frac{\sum_{j=1}^{i-1} e_j\lambda_j+d\lambda_i}{\Lambda}\right)$, then

$$\phi(M,u) = \begin{cases} M' & \text{if } e_i(M) = d \text{ and } M \stackrel{q_i}{\to} M' \\ M & \text{if } e_i(M) \neq d. \end{cases}$$

In that case, the blocking markings are not extremal anymore. Consider the example displayed in Figure 3, where all transitions have an enabling degree equal to two.

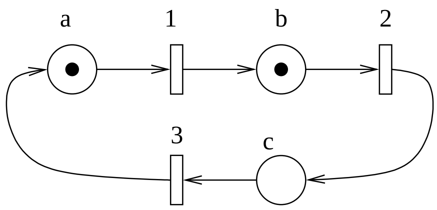

Figure 3: An event graph without self-loops with enabling degree two, for which PSA starting in blocking markings may not yield a stationary output.

The blocking markings are $(2,0,0),(0,2,0),(0,0,2)$. Now, consider the following sequence of firings (the corresponding rates are given in parenthesis) $q_3(\lambda_3), q_3(\lambda_3), q_2(\lambda_2), q_1(2\lambda_1), q_3(\lambda_3)$, where a firing with rate $2\lambda_1$ means that transition 1 must have an enabling of 2 in order to fire. Firing this sequence starting from all blocking states ends up in state $(1,1,0)$ while starting from the initial state given in the figure, the net reaches $(2,0,0)$. This happens with a positive probability for firing times exponentially distributed.

4.3.2 Free choice nets with self-loops

If the network is not an event graph, blocking states (associated with clusters this time) are not extremal either as shown by the example displayed in Figure 4.

The network in Figure 4 is a free choice net but not an event graph. We will now show that running a given firing sequence starting from the blocking states of all clusters may lead to a state which is not reached from some other states using the same firing sequence. Here, the blocking states are all of the form $(0,\ldots,0,2,0,\ldots,0)$. Let us fire the sequence 1, 1, 2, 2, 7, 5, 2, 4, 4, 2, 3, 7, 2, 4. All blocking states end in the state with two tokens in place a, while starting with the initial state given in Figure 4, the net reaches the state with one token in place a and one token in place b. This happens with a positive probability under exponential firing times assumptions.

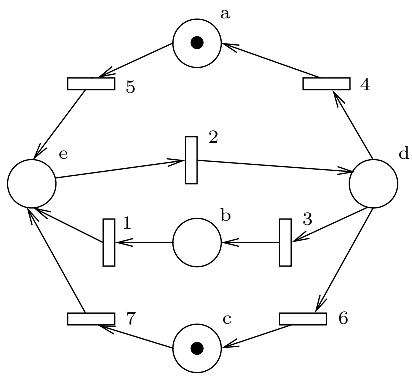

Figure 4: A free choice net for which PSA starting in blocking markings may not yield a stationary output

4.3.3 One-bounded free-choice nets

A more involved counterexample can also be found for one-bounded free choice nets (the previous example is bounded by two). Figure 5 represents a one-bounded free choice net where all blocking states may couple with positive probability in a marking which may be not steady state distributed. This example is rather large (20 transitions and 20 places) and the firing sequences involved in the counterexample are also long. This is the smallest example that we have been able to find. Consider the initial marking as drawn in Figure 5 and every blocking marking for that net. We will exhibit a sequence of transitions such that the trajectories from every blocking marking will join, but not the trajectory from the initial marking.

First, let us fire every possible transition once, except for transitions 1,2,3 and 4. As the only enabled transitions for the initial marking are transitions 3 and 4, that marking does not evolve, and the blocking markings of transitions 1, 2, 3 and 4 do not evolve either. To have a precise idea of the markings that are reached, let us consider a sequence that lets the transitions 1,2,3 or 4 to be the only enabled transition. First try to fire transitions 5, 6, 7, 8, 9, 10 in that order. While transition 1 and 2 are not fired, those transitions will not be enabled anymore for any future markings. Now, fire transitions 11, 12, 13, 14, 15, 16, 17. None of those transitions are enabled anymore, neither are transitions 18, 19 and 20. Moreover, places from a to ℓ are empty. Except for the initial marking and the blocking markings of transitions 3 and 4, the only enabled transitions are transitions 1 and 2 while transitions 1 and 2 are not fired, we have only three trajectories to consider.

In a next step, let us make the initial marking evolve, by firing transitions 3 and 4. Consider the firing sequence 3, 4, 5, 8, 9, 10, 18, 19. For the initial marking, the sequence that is effectively fired is 3, 4, 5, 8. For the blocking marking B_3, the sequence that is effectively fired is 3, 5, 9, 18, and there is a token in place i; for B_4, the sequence that is effectively fired is 4, 8, 10, 19, and there is a token in place f. While transitions 11 and 13 are not fired, the tokens in places f and i will not move and transition 15 cannot be fired for those trajectories from B_3 and B_4.

Now, fire transitions 6, 7, 9, 10, 12, 14. This sequence can be fired from the marking obtained from the initial marking, and transition 15 is now enabled. From the construction, this is the only reached marking (for all the trajectories computed) that can fire transition 15. So, if we fire transitions 15 and 20, there will be a token in place ℓ and this is the only marking that marks that place. If place ℓ is removed from the net, it remains live and one-bounded. Then there is a sequence of firing, that does not contain transition 20, such that every trajectory from all the blocking marking will couple, but not the trajectory from the initial marking, since place ℓ is marked for it.

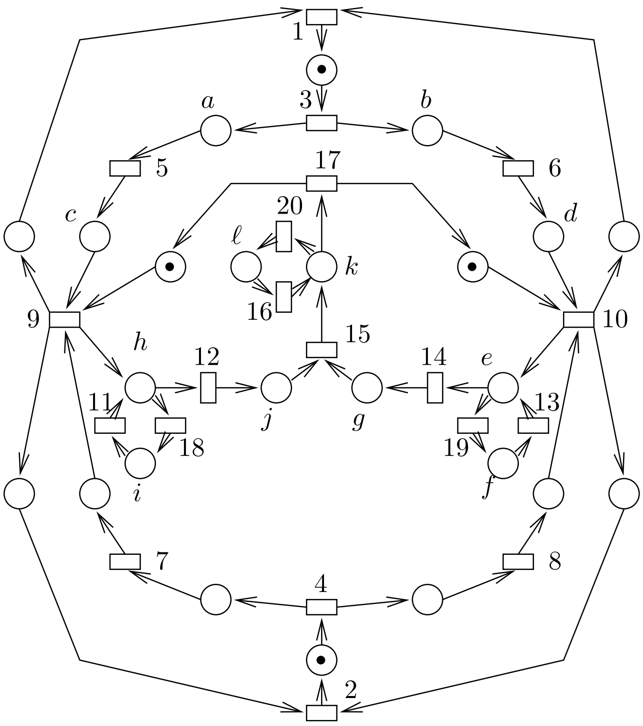

Figure 5: A 1-bounded free choice net for which PSA starting in blocking markings will not yield a stationary output.

5 Perfect Simulation using the (max,plus) algebra

A second method for perfect simulation of event graphs will be presented in the following. It is also based on backward coupling. However, unlike the Propp and Wilson Algorithm, this method does not require Markovian assumptions. This technique was used in the past (see for example [Mairesse, 1997], [Baccelli et al., 1992]) to prove the *existence* of stationary regimes for (max,plus) systems in a non constructive way. We will show here how it can also be used to get perfect *samplings* of the stationary regime of an event graph.

In the following we consider a bounded and live event graph, where all transitions have self-loops, under the following stochastic assumptions on the firing times:
(H_1): $\{\tau_q(n)\}_{n\in\mathbb{Z}}$ are i.i.d. and at least one is unbounded: $\exists t \in \mathcal{Q}$ s.t. $\forall x \in \mathbb{R}_+$, $\mathbb{P}(\tau_t(1) > x) > 0$ (for example the firing time of transition 1, τ_1, is unbounded).

Actually, more general conditions could be used (in particular without the unbounded support assumption), as explained in [Mairesse, 1997]. Everything done in this section is also true under these more general conditions, up to technical difficulties which are beyond the scope of this paper.

As shown in [Baccelli et al., 1992], the time evolution of event graphs with an initial marking bouned by one in all places can be written under the form of a (max,plus) linear equation [2] of size QQ in $\mathbb{R}_{\max} \stackrel{\text{def}}{=} (\mathbb{R} \cup \{-\infty\}, \vee, +)$. If $X_i(n)$ is the instant when the nth firing at transition q_i finishes, then there exists a sequence of (max,plus) matrices $\{A(n))\}_{n\in\mathbb{Z}} \in \mathbb{R}_{\max}^{QQ}$ such that

$$X(n) = X(n-1) \otimes A(n),$$

with

$$A_{ij}(n) \stackrel{\text{def}}{=} \bigvee_{c\in\mathcal{C}(i,j)} \sum_{q\in c} \tau_q(n),$$

where $\mathcal{C}(i,j)$ is the set of all paths from i to j containing no tokens except in their last place, containing one token, or $A_{ij}(n) = -\infty$ if no such path exists between i and j. For more on this, see [Baccelli et al., 1992], for example. Under assumptions (H_1), $(A(n))_{n\in\mathbb{Z}}$ is a sequence of i.i.d. matrices with a fixed support ($\mathbb{P}(A(1)_{ij} = -\infty) \in \{0, 1\}$).

The *profile* of a vector $v \in \mathbb{R}_{\max}^k$ is the non-negative vector $\gamma(v) \in \mathbb{R}_{\max}^k$ defined by $\gamma(v)_i = v_i - \min_{j=1}^k v_j$.

Definition 5. *A (max,plus) matrix $D \in \mathbb{R}_{\max}^{kk}$ is of rank 1 if all rows are equal up to an additive constant: $\forall i, j \quad \exists\, c_{ij}$ s.t. $c_{ij} + D_{i,.} = D_{j,.}$.*

[2] by definition, $(A \otimes B)_{ij} = \bigvee_k (A_{ik} + B_{kj})$

Alternatively, D is of rank 1 if and only if the profiles of all its rows are the same: $\forall i, j,\ \gamma(D_{.i}) = \gamma(D_{.j}) \Leftrightarrow \forall i, j,\ \gamma(D_{i.}) = \gamma(D_{j.})$.

Lemma 6. *If D is of rank one, then $\forall\, u, v \in \mathbb{R}^k$, $\gamma(u \otimes D) = \gamma(v \otimes D)$.*

Proof. For all i and j, using the definition of matrices of rank one, $(u \otimes D)_j = c_{ij} + (u \otimes D)_i$. If $i_0 =$ argmin$_i c_{1i}$, then the profile $\gamma(u \otimes D)_i = c_{i_o i}$ for all i. This does not depend on u. □

Let us now consider the sequence $(A(n))_{n\in\mathbb{Z}}$ of stochastic matrices corresponding to an event graph with Q transitions which is live and bounded. The algorithm for perfect simulation of the corresponding (max,plus) linear system is given in Algorithm 2.

Algorithm 2: Perfect Simulation Algorithm of (max,plus) linear systems

---

Data: A (max,plus) representation $X(n) = X(n-1) \otimes A(n)$ and a sequence of increasing integers $N_1, N_2, \ldots$

Result: A stationary profile of the (max,plus) system.

begin

  $m := 1$;

  repeat

    Compute $S_m := A(-N_m) \otimes \cdots \otimes A(0)$;

    $m := m + 1$;

  until S_m *is of rank one* ;

  $X(0) := X(-N_m) \otimes S_m$;

  return $\gamma(X(0))$

end

Under the foregoing assumption (H_1), it was proved in [Mairesse, 1997] that the profile of the system $X(n) = X(n-1) \otimes A(n)$ admits a stationary regime, *i.e.* that $\gamma(X_n)$ converges almost surely to a unique stationary profile γ_∞ (independent of the initial condition $X(0)$). The next theorem shows that Algorithm 2 provides samples of this stationary distribution.

Theorem 7. *If the (max,plus) perfect simulation algorithm 2 terminates, then its output has the distribution of the stationary profile of the (max,plus) system.*

Proof. Using Lemma 6, if the matrix S_m has rank one, and if X_∞ is a state with a stationary profile ($\gamma(X_\infty) = \gamma_\infty$ in distribution), then $X \otimes S_m = X_\infty \otimes S_m$ for all X. Since $\gamma(X_\infty)$ is stationary, so is $\gamma(X_\infty \otimes S_m) = \gamma(X \otimes S_m)$. □

The rest of this section is devoted to the proof that the (max,plus) perfect simulation algorithm terminates with probability one, under condition (H_1).

Lemma 8. *Under the foregoing assumption (H_1), the product $A(1) \otimes \cdots \otimes A(k)$ is of rank one with positive probability, as soon as $k > Q$.*

Proof. The main idea of the proof is to construct a set E of events over which the behavior of $A(1) \otimes \cdots \otimes A(k)$ is easy to control, namely remains close to a deterministic behavior $H \otimes \cdots \otimes H$ which will be of rank one.

For all i, j, let h_{ij} be any real number in the support of the random variable $A_{i,j}(1)$, verifying the following constraints. The set $\{h_{ij} \mid i \leq Q, j \leq Q\}$ does not satisfy any linear equation with coefficients in $\{1, -1\}$ involving h_{ij} whenever h_{ij} is in the interior of the support of $A_{i,j}(1)$. Also, h_{11} is such that $h_{11} > \sum_{(i,j) \neq (1,1)} Q(h_{ij} + 1)$. This is always possible because of assumption (H_1).

Now, let us consider the deterministic matrix $H = (h_{ij})$. The definition of (max,plus) matrices (see [Baccelli et al., 1992, Bouillard and Gaujal, 2001]), implies that the k-th power of H can be seen as weights of paths of length k in the graph G whose incidence matrix is H. More precisely, H^k_{ij} is the weight of the heaviest path of length k from node i to node j in the weighted graph G.

Since the weight h_{11} of the self-loop on node 1 is very high, the heaviest path of a length $k > Q$ consists in taking some heavy simple path from i to 1, loop on 1 as much as possible and take another heavy simple path from 1 to j. More precisely, for all i there exists a sequence $i_1 \ldots i_\ell$ and for all j there exists a sequence $j_1 \ldots j_r$ such that for all $k > Q$, $H^k_{ij} = h_{ii_1} + h_{i_1 i_2} + \cdots + h_{i_\ell 1} + (k - \ell - r - 2)h_{11} + h_{1j_1} + \cdots + h_{j_r j}$.

Using this form of the matrix H^k, it is straightforward to see that H^k is of rank one for all $k > Q$ (indeed the difference $H^k_{ij} - H^k_{\ell j}$ does not depend on j).

Now, let us consider the stochastic event $E_{k,\varepsilon}$ defined by

$$E_{k,\varepsilon} = \{\omega \mid A_{i,j}(n) \in [h_{ij} - \varepsilon, h_{ij} + \varepsilon] \ \forall n \leq k\},$$

Under event $E_{k,\varepsilon}$, if $k > Q$ and if ε is small enough, the product $A(1) \otimes \cdots \otimes A(k)$ is very close to H^k. Also, because the set $\{h_{ij} \mid i \leq Q, j \leq Q\}$ does not satisfy any linear equation in the interior of the support of A, two different simple paths $A(1)_{i_0 i_1} + \cdots + A(1)_{i_\ell i_{\ell+1}}$ and $A(1)_{j_0 j_1} + \cdots + A(1)_{j_r j_{r+1}}$ have weights which are equal or differ by more than ε, under event $E_{k,\varepsilon}$, as long as ε is small enough.

Therefore, $A(1) \otimes \cdots \otimes A(k)$ can be written as

$$
\begin{aligned}
(A(1) \otimes \cdots \otimes A(k))_{ij} = A(1)_{ii_1} + \cdots A(\ell+1)_{i_\ell 1} \\
+ A(\ell+2)_{11} + \cdots + A(k-r-1)_{11} \\
+ A(k-r)_{1j_1} + \cdots + A(k)_{j_r j},
\end{aligned}
$$

so that $(A(1) \otimes \cdots \otimes A(k))_{ij} - (A(1) \otimes \cdots \otimes A(k))_{i'j}$ does not depend on j so that $A(1) \otimes \cdots \otimes A(k)$ is also of rank one.

To finish the proof, it is enough to notice that under Conditions (H_1), $\mathbb{P}(E_{k,\varepsilon}) > 0$ for all $k \in \mathbb{N}$ and all $\varepsilon > 0$. □

Using Lemma 8 and the Borel-Cantelli theorem, it is now direct to show the following result.

Theorem 9. *Under assumptions (H_1), the (max,plus) perfect simulation algorithm terminates with probability one.*

¿From a stationary profile γ_∞, it is possible to get a stationary marking of the event graph by appending the following steps in the (max,plus) simulation algorithm. The output is a marking distributed according to the stationary distribution.

Algorithm 3: Generating a stationary marking

Data: Sample of a real non-negative r. v. d independent of everything

Result: Stationary marking

begin
- $k := 0$; $M := M_0$ (initial marking);
- $X(0) := \gamma(X_\infty)$ (stationary profile);
- **repeat**
  - $X(k) := X(k-1) \otimes A(k)$;
  - $k := k + 1$;
- **until** $X_i(k) > \max_j X_j(0) + d, \ \forall i$;
- **forall** *transition* s_i **do**
  - $n_i := \max\{n \mid X_i(n) < \max_j X_j(0) + d\}$;
  - Update M firing n_i times transition q_i;
- **return** M

end

6 Comparison of the two methods

We have implemented the two methods presented above to simulate a simple event graph. Although the (max,plus) algorithm is more general (does not need exponential firing times), we have used exponential firing times to be able to compare both methods over the same example. The programs are both written in Caml, using in both cases the most efficient methods known to us. In particular, in the Markovian case, the sequence of integers N_m used at each step is $N_m = 2^m$ which was proved optimal in average for the Markov chain algorithm in [Propp and Wilson, 1996]. The Markov chain algorithm also uses an aliasing technique that enables one to compute $\phi(X, U)$ in almost constant time for any $U \in [0, 1]$. This technique replaces the real-valued random variable U by a couple (U, V) where U is real-valued, uniformly distributed over $[0, 1/Q]$ and V is integer valued, uniformly distributed over $\{1, \ldots, Q\}$. It was first developed in [Walker, 1974] and has been used in [Vincent and Marchand, 2004], for perfect simulation.

In the experiments given below, the (max,plus) algorithm computes a stationary profile. The additional matrix products needed to get a stationary state are not included. They should increase the simulation time by a rather small quantity.

The event graph used in the simulations is a simple circuit made of K transitions (and K places) and W tokens in total. The (max,plus) representation of such a network uses a matrix $A(n)$ with size $Q = \max(K, W)$. The total number of states is $\binom{K+W-1}{K-1}$. In the experiments, $K = 40$ and W ranges from 1 to 80, so that the number of states goes up to $3.819\ 10^{31}$. This prevents the use of direct numerical approaches to compute the stationary distribution.

Figure 6 displays the number of iterations for both algorithms, while Figure 7 displays the total simulation time. Each point is the average of many simulations (enough to guarantee confidence intervals smaller than 1 % for the empirical mean).

The number of iterations before coupling is much smaller for the (max,plus) case. This is because one step in the (max,plus) algorithm corresponds to firing all the transitions in the net once, while one step in Algorithm 1 is the firing of a single transition. However, the simulation time is much larger for the (max,plus) algorithm because one step corresponds to the product of two large matrix (of size K), involving $O(K^3)$ elementary operations. One can notice that the time complexity of the (max,plus) algorithm starts to increase rapidly when the number of tokens W becomes larger than the number of transitions K. One explanation is that from that point on, the size of the matrices starts to increase from K to W. The same kind of behavior (fewer iteration but larger simulation time) has been observed when the number of transitions K changes. The

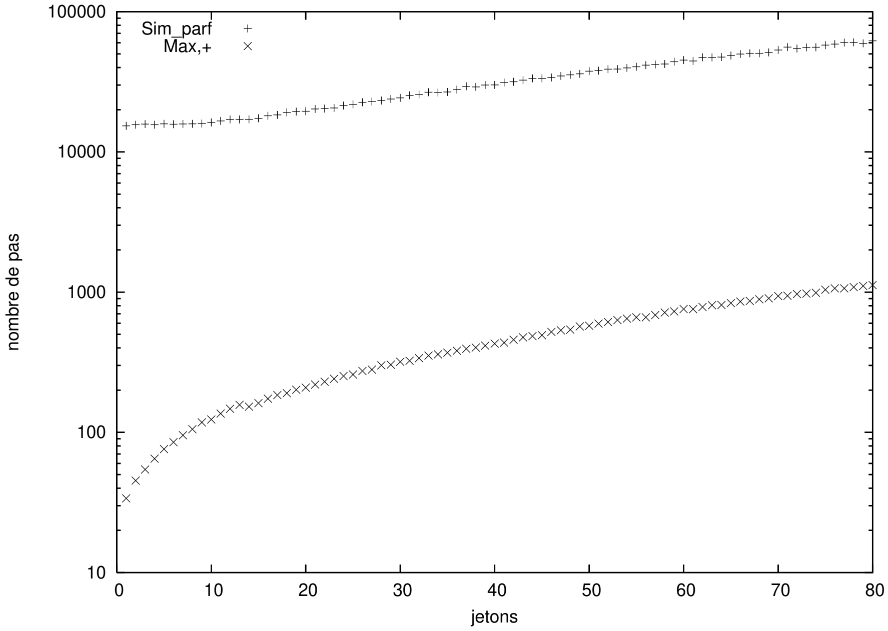

Figure 6: Average number of iterations for a circuit with $K = 40$ transitions when the number of tokens varies.

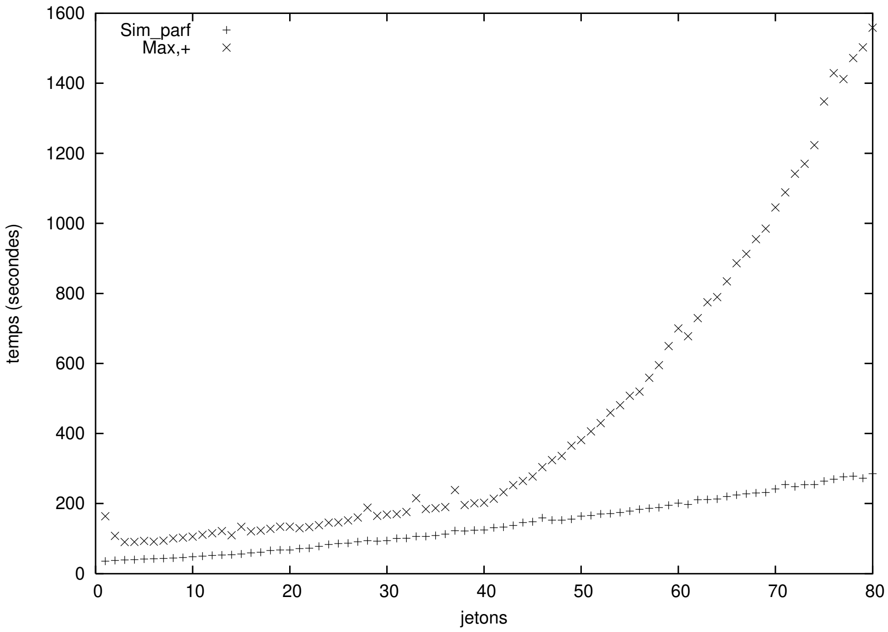

Figure 7: Average simulation time for a circuit with $K = 40$ transitions when the number of tokens varies.

corresponding curves are similar to those in Figures 6 and 7 but are not reported here.

While the Markov chain method is faster, the (max,plus) one is more general in terms of firing distributions. Futhermore, the (max,plus) method can also be used to simulate the stationary distribution of (max,plus) systems where the support of the matrices is not fixed as shown in the following section.

7 Several extensions: heaps of pieces and free-choice nets

7.1 Heaps of pieces

Consider a finite set $\mathcal{C}$ of *computing resources* (or columns) and a finite set $\mathcal{A}$ of *tasks* (or pieces).

Here, we will consider arbitrary pieces (described below). For $a \in \mathcal{A}$, $C(a)$ is the set of resources used to execute task a. In order to execute task a, all ressources r in $C(a)$ are requested at times $b_r(a)$ and they are released at different instants $e_r(a)$. The time elapsed between those two instants is the execution time of a on ressource r: $h_r(a)$.

It is shown in [Gaubert and Mairesse, 1999] that the execution of the sequence of tasks $a_0 \dots a_n$ under the asap policy forms a heap of pieces as illutrated by Figure 8.

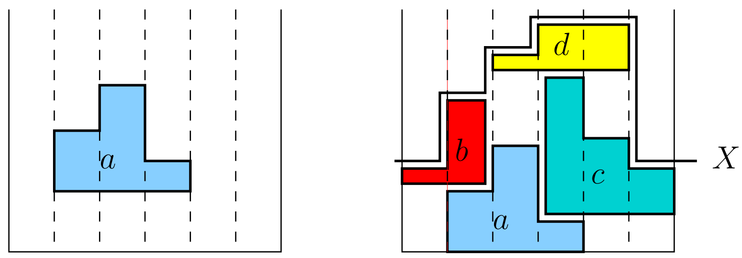

Figure 8: A piece corresponding to a task (a), the heap corresponding to the execution of a sequence of tasks, *abcd* and the corresponding execution makespan over all ressources, X.

The total execution time (makespan) over all the ressources is given by the upper level of the heap, the vector

$$X(n) = X(0) \otimes T(a_1) \otimes \cdots T(a_n),$$

where X(0) = (0,. . . ,0) and for all $a \in \mathcal{A}$, the matrix

$$T(a)_{sr} = \begin{cases} 0 & \text{if } s = r, r \notin C(a) \\ e_r(a) - b_s(a) & \text{if } s, r \in C(a) \\ -\infty & \text{otherwise.} \end{cases}$$

For stochastic heaps, there are two types of randomness. First, for a given a, $b_r(a)$ and $e_r(a)$ are sequences of i.i.d. random variables for all ressources $r \in C(a)$, while the support $C(a)$ is fixed. Second, at each step, one chooses which task to execute next by picking a in $\mathcal{A}$ according to an i.i.d. Bernoulli distribution (with probability $p_a > 0$).

Here are some technical assumptions on the set of task matrices $T(a), a \in \mathcal{A}$.

$$(H_2) : \begin{cases} \forall a \in \mathcal{A}, \forall r \in C(a),\ \{b_r(a)(n), e_r(a)\}_{n\in\mathbb{N}} \text{ are i.i.d. with } e_r(a) \geq b_r(a) \text{ and} \\ \qquad\qquad e_r(a) - b_r(a) \text{ is strictly positive for al least one } r \\ \forall\, r \in \mathcal{C},\ \exists\, a \in \mathcal{A} \text{ such that } r \in C(a) \\ \forall\, r, s \in \mathcal{C}, \exists\, a_1 \dots a_k \in \mathcal{A}, \quad \exists\, r_0 (=r), r_1, \dots, r_k (=s) \text{ such that } r_{i-1}, r_i \in C(a_i). \end{cases}$$

In words, the last two assumptions correspond to the fact that all resources ae used and that the heap cannot be split into two independent sub-heaps.

Theorem 10 ([Gaubert and Mairesse, 1999])**.** *Under assumptions (H_2), the sequence $\gamma(X_n)$ converges in law to a unique stationary profile.*

The goal of the rest of this section is to provide an algorithm to sample this stationary profile with no bias.

Algorithm 4: Perfect sampling of the stationary distribution of a heap of pieces

---

Data: An infinite Bernoulli sequence in $\mathcal{A}$, $a_1, \dots, a_n, \dots$, matrices $T(a_1), \dots T(a_n), \dots$ and a sequence of increasing integers $N_1, N_2, \dots$.

begin
- $m := 1$;
- **repeat**
  - Compute $S_m := T(a_{-N_m}) \otimes \cdots \otimes T(a_0)$;
  - $m := m + 1$;
- **until** S_m *is of rank one* ;
- $X(0) := X(-N_m) \otimes S_m$;
- **return** $\gamma(X(0))$

end

Theorem 11. *Algorithm 4 terminates in finite time w.p. 1 and its output $\gamma(X(0))$ has the stationary distribution of the heap of pieces.*

Proof. Consider a finite sequence $a_1 \dots a_n$ of pieces such that $C(a_1) \cup \cdots \cup C(a_n) = \mathcal{C}$ and such that $C(a_i) \cap C(a_{i+1}) \neq \emptyset$ for all $1 \leq i \leq n-1$. Such a sequence exists because of assumption (H_2).

For some $B > 0$, consider the event E_B defined as the set of all pieces such that the differences between the starting times are bounded: $|e_r(a) - e_s(a)| < B$ for all r, s and a. In the following, all pieces belong to the stochastic event E_B.

The next point is to construct a sequence of pieces such that the upper profile of the heap once this sequence is stacked does not depend of the initial heap (the corresponding matrix will be of rank one).

We first consider the sequence $s_1 = a_1 \ldots a_n$. Once this sequence of pieces is stacked, the last piece, a_n rests over some solid ground, whatever the initial state of the heap.

Next, let us stack the sequence $s_2 = a_n a_{n-1}^{d_{n-1}} a_{n-2}^{d_{n-2}} \ldots a_1^{d_1}$ where the number of times d_i a given piece a_i is stacked upon itself is defined as follows: the height of the heap over all columns in $C(a_i)$ after $a_n a_{n-1}^{d_{n-1}} a_{n-2}^{d_{n-2}} \ldots a_i^{d_i}$ have been stacked over a flat initial state, is larger than the highest column before $a_i^{d_i}$ has been stacked, plus B. This is possible because a_i belongs to E_B and d_i is a well defined finite random variable which only depends on the sequences $h_r(a_j)(k)$ for all $j \geq i$ and all $k \leq d_j$.

Now, once the sequence s_2 hase been stacked, it should be clear that the columns in $C(a_1)$ are the highest of the heap, whatever the initial state, by a margin larger than B.

¿From that point one, if we stack the sequence $s_1 = a_1 \ldots a_n$, again, then all pieces will get stacked above the height of the columns in $C(a_1)$, by event E_B. The final heap has an upper profile that does not depend on the initial state of the heap, before the sequence $s_1 s_2 s_1$ has been stacked. This means that the corresponding matrix

$$T(a_1) \otimes \cdots \otimes T(a_n) \otimes T(a_n) \otimes T(a_{n-1})^{d_{n-1}} \otimes \cdots \otimes T(a_1)^{d_1} \otimes T(a_1) \otimes \cdots \otimes T(a_n)$$

is a matrix of rank one. Since the probability of each piece is positive and since the selection process is i.i.d., The probability that the sequence $s_1 s_2 s_1$ occurs is also positive. Using Borel-Cantelli's Lemma, such a sequence will occur w.p. 1 in any infinite sequence. Now, once the matrix is of rank one, any additional product remains of rank one. Therefore, the algorithm stops w.p. 1, at step, say m. As for the output, let X_∞ be a heap with a stationary profile. Then $\gamma(X_0 \otimes S_m) = \gamma(X_\infty \otimes S_m)$ because B_m is of rank 1, and $\gamma(X_\infty \otimes S_m)$ has the same distribution as $\gamma(X_\infty)$ because X_∞ is stationary. $\square$

7.2 Safe free choice nets

In this section, we consider one-bounded (also called safe) free choice nets (SFCNets), with i.i.d. stochastic positive firing times having arbitrary distributions, under *Bernoulli routing* (in all places, the tokens are routed to output transitions according to fixed positive probabilities: c_{pq} is the probability to route a token in place p to transition $q \in p^\bullet$). The behavior of such SFCNets can be represented as heaps of pieces (see for example [Gaubert and Mairesse, 1999]).

Here, a piece corresponds to a transition and the resources are the places. The resources $C(q)$ used by transition q, are all the input and output places of the transition.

More precisely, $\mathcal{C} = \mathcal{P}, \mathcal{A} = \mathcal{Q}$ and for any transition q with firing time τ_q, $C(q) = {}^\bullet q \cup q^\bullet$ and

$$T(q)_{ij} = \begin{cases} 0 & \text{if } i = j, j \notin C(q) \\ \tau_q & \text{if } i \in C(q), j \in q^\bullet \\ 0 & \text{if } i \in C(q), j \in {}^\bullet q \setminus q^\bullet \\ -\infty & \text{otherwise.} \end{cases}$$

Note that the piece corresponding to a transition has a flat bottom. If the firing times are i.i.d. in all transitions and if the Petri net is strongly connected, then assumptions (H_2) are verified for the corresponding heap of pieces. However, the behavior of SFCNets has one main difference with classical (free) heaps: the sequence of pieces being stacked cannot be arbitrary, only sequences of pieces corresponding to enabled transitions can possibly be stacked. One way to deal with this restriction is to construct a new heap of pieces by considering the tensor products of the (max,plus) matrices $T(t)$ corresponding to the pieces with a matrix modeling the reachability graph of the marking of the free choice net. Doing this makes Theorem 11 applicable so that Algorithm 4 terminates in finite time w.p. 1 and its output $\gamma(X(0))$ has the stationary distribution of the heap of pieces. However, this approach has a major drawback: the size of the matrices becomes $O(P|\mathcal{R}(M_0)|)$ so one loop of the simulation algorithm takes $O(P^3|\mathcal{R}(M_0)|^3)$ steps and $O(P^2|\mathcal{R}(M_0)|^2)$ memory slots. This can be too large to be handled by a computer even for Petri nets of moderate sizes, since $\mathcal{R}(M_0)$ may be exponential in the size of the net.

Here, we propose an alternative approach, which is to consider only the initial free heap of pieces (without using the tensor product) and at each step, check whether a given new piece can be stacked. The overall complexity of one loop of the simulation algorithm (Algorithm 5) will be in $O(P^3|\mathcal{R}(M_0)|)$ steps and $O(P|\mathcal{R}(M_0)|)$ memory slots, which makes perfect simulation of moderate size SFCNets possible. The main ingredient of this approach is the following lemma, relating the profile of the heap of pieces to the corresponding marking in the original SFCNets.

Lemma 12. *Consider a SFCNet and the corresponding heap of pieces. After firing transitions* $q_1 \dots q_n$*, the heap has reached the state* $X(n) = X(0) \otimes T(q_1) \otimes \dots \otimes T(q_k)$*. Then the marking in the SFNet* $Mark(X_n)$ *can be recovered from the profile* $\gamma(X(n))$*.*

Proof. A SFCNet is covered by one-bounded S-components [Desel and Esparza, 1995]. For each S-component, when firing transitions $q_1 \dots q_n$, the latest event within the S-component has moved its only token to some place p. Therefore, this place corresponds to the highest value of $X(n)$ among all the places in the S-component. Knowing the position of the tokens in all the S-components covering the SFCNet determines the marking. $\square$

Remark that a profile that corresponds to a marking can be easily found: the height of the heap at a place p is 0 if there is one token in that place and $-\infty$ otherwise.

Algorithm 5: PSA of the stationary profile of the heap of a SFCNet

---

Data: (P, Q, τ, C) a SFCNet, Bernoulli routings $\{c_{pt}\}$, a sequence $(q_1, \tau_1), (q_2, \tau_2) \dots$ of transitions with their firing time and $N_1, N_2, \dots$ an increasing sequence of integers.

begin
- $m := 1$;
- **forall** *Marking* $M \in \mathcal{R}(M_0)$ **do**
  - Compute a heap $X_M^{(0)}$ such that $Mark(X_M^{(0)}) = M$.
- **repeat**
  - **foreach** *reachable marking* M **do**
    - $S_M := Id$; $X_M := X_M^{(0)}$;
    - **foreach** *i from 1 to* N_m **do**
      - **if** q_{N_m-i+1} *is enabled in* $Mark(X_M)$ **then**
        - $S_M := S_M \otimes T(q_{N_m-i+1}, \tau_{N_m-i+1})$;
        - $X_M := X_M \otimes S_M$;
  - $m := m + 1$;
- **until** S_M *is of rank one and* $Mark(X_M)$ *is the same, for all markings* $M \in \mathcal{R}(M_0)$. ;
- **return** $\gamma(X_M)$, *for one* $M \in \mathcal{R}(M_0)$.

end

In Algorithm 5, a transition of the sequence is chosen to be q with probability $c_{pq}/|\Gamma|$, where p and q belong to the same cluster.

Here is the main result of this section, based on the following set of assumptions,

$$(H_3) : \begin{cases} \text{The SFCNet is live and strongly connected} \\ \text{The firing times } \tau_q(n) \text{ are i.i.d.} \\ \text{At least one is unbounded: } \exists q \in \mathcal{Q} \text{ s.t. } \forall x \in \mathbb{R}_+,\ \mathbb{P}(\tau_q(1) > x) > 0. \end{cases}$$

Theorem 13. *Under assumptions* (H_3),

i- Algorithm 5 terminates in finite time w.p.1.

ii- Its output is sampled according to the stationary distribution of the heap.

Proof. Before proving termination, one must convince oneself that each trajectory simulated by Algorithm 5 is valid. This is because for any given initial marking M, the choice of the next transition to fire is such that it is either invalid (in which case nothing happens) or follows the good Bernoulli proportions.

For claim *i-*, the termination of the algorithm can be proved in two steps. First, we need to show the convergence of the marking. Using Lemma 12, the marking associated with heap $X(n)$ is unique so that this part can be proved in the same way as Theorem 1.

Next, one needs to show that the profiles also converge. One cannot use the results of the previous section, because every heap is not admissible, so a sequence $s_1 s_2 s_1$ constructed in the proof on Theorem 11 may not be enabled. Suppose that while running Algorithm 5, a common marking M_c has been reached, starting from all initial markings. From that point on, one only needs to exhibit a firing sequence such that the corresponding pieces form a matrix of rank one. Consider a transition q and B_q the blocking marking of q. The marking B_q is reachable in finite time almost surely (and this does not depend on the timings of the transitions, but only on the firing sequence), so one can suppose that the common marking M_c that is reached after the coupling on the markings is in fact B_q. Now consider a firing sequence $\sigma = q_1 \ldots q_n$ that is admissible from B_q such that the firing time of q_n has an unbounded support, σ contains every transition at least once and $B_q \xrightarrow{\sigma} B_q$. Such a sequence exists, is finite and corresponds to the (max,plus) matrix $A(1) = T(q_1) \otimes \ldots \otimes T(q_n)$. The matrix $A(1)$ is a stochastic matrix with a fixed support and is irreducible. Under assumptions (H_3), the construction used in the proof of Lemma 8 can be adpated for matrix $A(1)$ (replacing the cycle of length one over the unbounded transition by a cycle of finite length, involving the transition with an unbounded support). Therefore, the matrix $A(1) \otimes A(2) \otimes \cdots \otimes A(P)$ is of rank one with positive probability and corresponds to the firing sequence σ^P (which is admissible). The sequence σ^P occurs with positive probability, because B_q will be reached infinitely often. So, using Borel-Cantelli's Theorem, the algorithm yields a matrix of rank one with probability one.

As for *ii-*, the proof is similar to the proof of Theorem 7. □

To recover a marking for the SFCNet, with a stationary distribution, one simply needs to run an algorithm similar to the one given at the end of Section 5.

Unlike for free heaps of pieces, this PSA needs to compute one trajectories per marking in $\mathcal{R}(M_0)$. Therefore, it may only be used for nets of moderate size. However, note that computing the stationary distribution is already very hard with two or three nodes when the distributions of the firing times are not exponential. Therefore, our perfect sampling technique remains interesting for small nets.

7.3 Application in one example

The proof of Theorem 13 constructs a coupling sequence which has a very small probability to occur. This may suggest that the coupling time of Algorithm 5 may actually be extremely large, jeopardizing its applicability. We have checked that this is not the case over the net given in Figure 9. We also performed several simulations over this net, using several routings policies and several distributions for the timings to get an idea on the dependence of the coupling time on those parameters.

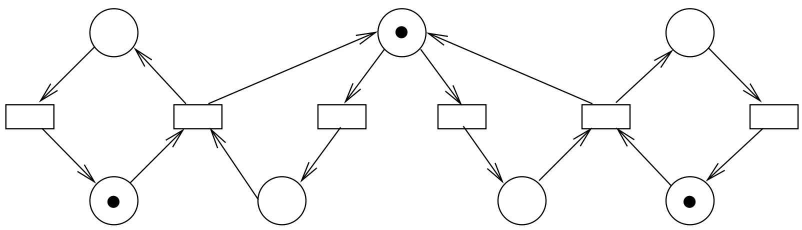

Figure 9: An example of a SFCNet

Three types of parameters are used. For every experiment, Algorithm 5 has been run 1000 times and the empirical average of the coupling time (number of iterations) is reported.

First, the average value of the firing times ranges from 1 to 10 (all of them being exponential). In that case, the coupling time of Algorithm 5 remains in the interval $[96, 99]$.

Second, the average firing time is fixed to one for all transtitions, but the distributions of the firing times in all 6 transitions is selected among exponential (Exp) determistic (Det) and Erlan 4 (Erl) or a mixture among them (Mix). As seen in Figure 10, the distribution has a very marginal effect on the coupling time.

Finally, the routing parameter in the central place ranges from 0 to 1. In that case, the coupling time is varying too, as shown in Figure 10. Note that the number of iterations still remains managable. The fact that it increases when almost all tokens are sent to the right (or to the left) comes from the fact that strong connectedness tends to disappear, making rank one profiles hard to reach.

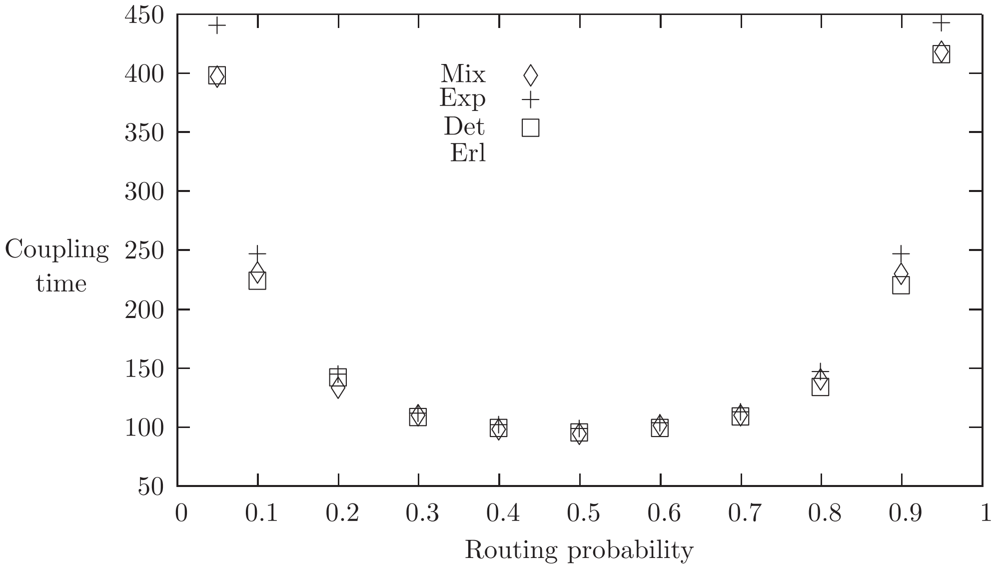

Figure 10: Dependence of the coupling time on the routing and on the distributions of the firing times

7.4 Extensions

The perfect sampling technique developped for SFCNets can be extended to more general cases.

- If firing conflicts are solved using the *race policy* (as in Section 3) this is also amenable to perfect sampling using Algorithm 5 because the race policy can be modeled by Bernoulli routings by modifying the distribution of the firing times according to the following transformation. The distribution of the new firing time τ'_q of transition q becomes

$$\mathbb{P}(\tau'_q \leq x) = \mathbb{P}(\tau_q \leq x | \forall y \in ({}^\bullet q)^\bullet, \tau_y \leq \tau_q).$$

This shows the importance of removing all assumptions on the firing times: a FCNet using the race policy is equal in distribution to a FCNet with Bernouilli routings which firing times have rather complex i.i.d. distributions.

- I.i.d. assumptions for firing times can also be replaced by more general stationary and ergodic assumptions. However this extension introduces several technical difficulties similar to those in [Baccelli et al., 1996], which are beyond the scope of this paper.

Acknowledgment. The authors would like to thank Jean Mairesse for his precious advices.

References

[Baccelli et al., 1996] Baccelli, F., Foss, S., and Mairesse, J. (1996). Stationary ergodic jackson networks: Results and counter-examples. *Stochastic networks, Oxford Univ. Press*, pages 281–307.

[Baccelli et al., 1992] Baccelli, F., Gohen, G., Olsder, G., and Quadrat., J.-P. (1992). *Synchronization and Linearity*. Wiley.

[Baccelli et al., 1995] Baccelli, F., Jean-Marie, A., and Mitrani, I., editors (1995). *Quantitative Methods in Parallel Systems*. Basic Research Series. Springer Verlag.

[Baccelli and Mairesse, 1998] Baccelli, F. and Mairesse, J. (1998). *Idempotency*, chapter Ergodic Theory of Stochastic Operators and Discrete Event Networks, pages 171–208. Publications of the Isaac Newton Institute. Cambridge University Press.

[Bouillard and Gaujal, 2001] Bouillard, A. and Gaujal, B. (2001). Coupling time of a (max,plus) matrix. In *Workshop on Max-Plus Algebras and Their Applications to Discrete-event Systems, Theoretical Computer Science, and Optimization*, Prague. IFAC.

[Bouillard and Gaujal, 2006] Bouillard, A. and Gaujal, B. (2006). Backward coupling in petri nets. In *Valuetools*, Pisa, Italy.

[Desel and Esparza, 1995] Desel, J. and Esparza, J. (1995). *Free Choice Petri Nets*. Cambridge Tracts in Theorical Computer Science.

[Foss et al., 1998] Foss, S., Tweedy, R., and Corcoran, J. (1998). Simulating the invariant measures of markov chains using backward coupling at regeneration time. *Probability in the Engineering and Informational Sciences*, 12:303–320.

[Gaubert and Mairesse, 1999] Gaubert, S. and Mairesse, J. (1999). Modeling and analysis of timed Petri nets using heaps of pieces. *IEEE Trans. Autom. Control*, 44(4):683–697.

[Gaujal et al., 2003] Gaujal, B., S.Haar, and Mairesse, J. (2003). Blocking a transition in a free choice net, and what it tells about its throughput. *Journal of Computer and System Sciences*, 66(3):515–548.

[Häggström, 2002] Häggström, O. (2002). *Finite Markov chains and Algorithmic Applications*, volume 52 of *Student texts*. Cambridge University Press.

[Mairesse, 1997] Mairesse, J. (1997). Products of irreducible random matrices in the (max,+) algebra. *Adv. Applied Probability*, 29(2):444–477.

[Propp and Wilson, 1996] Propp, D. and Wilson, J. (1996). Exact sampling with coupled Markov chains and application to statistical mechanics. *Random Structures and Algorithms*, 9(1):223–252.

[Vincent and Marchand, 2004] Vincent, J.-M. and Marchand, C. (2004). On the exact simulation of functionals of stationary Markov chains. *Linear Algebra and Its Applications*, 386:285–310.

[Walker, 1974] Walker, A. (1974). An efficient method for generating random variables with general distributions. *ACM Trans. Math. Software*, pages 253–256.