Backward Coupling in Petri nets

Anne BOUILLARD
Institute of Communication engineering
National Tsinghua University,
101, Kuangfu Rd, S. 2, Hsinchu, 300, Taiwan.
Email: Anne.Bouillard@ens-lyon.fr

Bruno GAUJAL
INRIA
Lab. ID-IMAG (CNRS, INPG, INRIA, UJF)
51, Av. J. Kunztmann, Montbonnot, France
Email: Bruno.Gaujal@imag.fr

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. Next, we 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.**

I. 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 [2]. 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 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 ([13]) 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 open networks with blocking and rejection (see [14]). 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 one transition (no firing is allowed) and let the system evolve until a deadlock is reached. Doing this, one gets the Q blocking states of the network.

A second method for perfect sampling, based on a (max,plus) representation of the dynamics of network, 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 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 [7], [12], [3]. This theory has been used mainly to prove *existence* theorems in full generality [3]. 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.

II. 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

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

with X_n the state of the chain at time n and $\{u_n\}_{n\in\mathbb{N}}$ an independent and identically distributed sequence of real random variables, uniformly distributed over $[0,1]$. The transition function $\phi : \mathcal{S}\times\mathcal{E}\to\mathcal{S}$ 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.

Let $\phi^n : S\times\mathcal{E}^n\to\mathcal{S}$ 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 note

$$\phi^n\left(A, u_{1\to n}\right) = \left\{\phi^n\left(s,u_{1\to n}\right), s\in A\right\}.$$

In the following, $|X|$ denotes the size of set X.

Theorem 1 ([13]): Let ϕ be a transition function on $\mathcal{S}\times\mathcal{E}$. 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 [8] In particular, the main result of the backward scheme is the following theorem, justifying the name “Perfect Sampling”.

Theorem 2 ([13]): 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) (1) sampling the steady state can be constructed. The perfect simulation algorithm that we will be using in Sections II, III and IV is given in Figure 1. Note that the same variables $U_0, U_{-1}, U_{-2},\cdots$ are used for all the simulations.

```
Input A recurrent representation $\phi$ 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 $[0,1]$.
$m := 1$
repeat
  for all 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 $U_{-N_m+1},\cdots,U_0$.
  end for
  $m := m+1$
until all simulations end up in the same state $(X)$
Output $X$
```

Fig. 1. Perfect Simulation Algorithm (PSA) of Markov chains

Note that a given Markov chain 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 [11] 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) so that it is well suited (or not) for PSA.

III. Sampling of Markovian Free-Choice Petri Nets

A Stochastic Free-Choice *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 $\mathcal{F}\subset(\mathcal{P}\times\mathcal{Q})\cup(\mathcal{Q}\times\mathcal{P})$ 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 from M_0 is $\mathcal{R}(M_0)$. For a node $x\in\mathcal{P}\cup\mathcal{Q}$, we denote by ${}^\bullet x$ the set of its predecessors and by $x^\bullet$ the set of its successors. 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 firing times in the transitions $\tau = (\tau_q)_{q\in\mathcal{Q}}$ are random sequences $\tau_q = (\tau_q(n))_{n\in\mathbb{N}^*}$ of i.i.d. variables with finite expectations $(E(\tau_q(1))<\infty)$.

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 boundedness assumption implies that the net is closed. To model the mono-server case, we assume that all transitions have self loops and firings occur using the *race policy* (*i.e* if two transitions are in the same cluster, the one with the smallest firing time fires the tokens in their common input places). For more details and precise definitions on Petri nets, we refer to [6].

In the following, 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 form of a finite continuous time Markov chain which infinitesimal generator is $W = (W_{M_1,M_2})_{M_1,M_2 \in \mathcal{R}}$ 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

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

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,

$$\text{if } u \in \left[\frac{\sum_{j=1}^{i-1} \lambda_j}{\Lambda}, \frac{\sum_{j=1}^{i} \lambda_j}{\Lambda}\right), \text{ then}$$

$$\phi(M,u) = \begin{cases} M' & \text{if } M \xrightarrow{q_i} M' \\ M & \text{if } q_i \text{ is not enabled in } M. \end{cases}$$

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 and is irreducible because the network does not contain any deadlock. 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 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 [14] 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 (server) a is the state reached eventually, after blocking transition a.

It has been proved in [10] that for the class of bounded Petri nets used here with no deadlocks, such states are unique, no firing is possible under B_a except at transition a, and that B_a is reachable from any state in $\mathcal{R}$ without ever using transition a. For more on blocking states (in particular on their regeneration properties, see [10]).

Here is 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 exogenous sequence, but starting from M_2, we get $\phi^\ell(M_2,, u_1 \dots u_\ell) = M_3$ for some M_3. 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 serves so that no state change happens starting in state B_a.

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

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 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 only works for event graphs.

IV. Sampling of Event graphs

Here, we consider a Markovian event graph which is bounded and has no deadlock. Since every place has a single input transition (q) and a single output transition (s), 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, $\{B_a, a \text{ transition}\}$, will provide a perfect sampling when coupling occurs.

A. Blocking states

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 [14]), 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 firing, s a transition and M a state of the net, we denote by $N_s(\sigma, M)$ is the number of times firing actually occurs at s, when starting from M and trying to proceed through the sequence of firing σ in that order (after k steps, if firing σ_k is allowed, then it is performed, otherwise nothing happens and the next firing is tried). This is also called *firing σ from M*.

Let M be a state under which only two transitions are enabled, say a and b. Let us denote by σ_a the shortest sequence of firings that leads from M to B_a, not including a and by σ_b the shortest sequence of firings that leads from M to B_b, not including b. One knows that two such sequences exist according to [10].

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

$$\begin{aligned} N_s(\sigma, M) \quad &= \quad N_s(\sigma_a, M) + N_s(\sigma, B_a) \\ &\qquad \wedge\ N_s(\sigma_b, M) + N_s(\sigma, B_b). \end{aligned}$$

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 of 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 and the first firing in σ_b must be a for similar reason. Let s be the first firing in σ_a common with σ_b. Since s was not allowed under M, then some other firings must have brought packets in the incoming buffers of s. But these firings must have occurred in both σ_a and σ_b, contradicting the fact that s 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 $|\sigma| = n + 1$. Let $\sigma = \sigma' s$.

If

$$\min_{t \in {}^{\bullet}s} (N_t(\sigma', M) + M(t, s)) > N_s(\sigma', M), \qquad (3)$$

then $N_s(\sigma, M) = N_s(\sigma', M) + 1$ else $N_s(\sigma, M) = N_s(\sigma', M)$.

By induction,

$$\begin{aligned} & \min_t (N_t(\sigma', M) + M(t, s)) \\ = \quad & \min_t (N_t(\sigma', B_a) + N_t(\sigma_a, M) + M(t, s) \\ & \wedge N_t(\sigma', B_b) + N_t(\sigma_b, M) + M(t, s)) \\ = \quad & \min_t (N_t(\sigma', B_a) + N_s(\sigma_a, M) + B_a(t, s) \\ & \wedge N_t(\sigma', B_b) + N_s(\sigma_b, M) + B_b(t, s)) \\ = \quad & N_s(\sigma_a, M) + \min_t (N_t(\sigma', B_a) + B_a(t, s)) \\ & \wedge N_s(\sigma_b, M) + \min_t (N_t(\sigma', B_b) + B_b(t, s)). \end{aligned}$$

Using this equation, it should be clear that $N_s(\sigma, M)$ increases by one if and only if the minimum of $N_s(\sigma, B_a)$ and $N_s(\sigma, B_b)$ increases by one. ■

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,

$$\begin{aligned} N_s(\sigma, M) &= N_s(\sigma_1, M) + N_s(\sigma, B_1) \quad (4) \\ &\wedge N_s(\sigma_2, M) + N_s(\sigma, B_2) \quad (5) \end{aligned}$$

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, and then by using Theorem 1. Let us assume that coupling from the past occurs for all blocking states after k steps. The corresponding sequence of firings is denoted σ and the coupling state is denoted by C.

Let denote by M 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 running 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 running the sequence σ.

Lemma 4, says that after running σ from M or from B_1 (and B_2), a new firing occurs in both cases together or for none of them. This is true for any new sequence of firings. This means that all three states are equal after running σ. This common state must be C. ■

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

B. Counter examples 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.

1) Event graphs without self-loops: While the general simulation scheme can be readily adapted for transitions with no self-loops (taking into account the enabling degree of a transition) the blocking markings are not extremal anymore. Consider the example displayed in Figure 2.

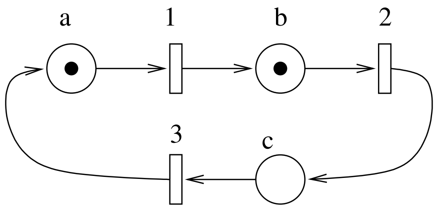

Fig. 2. An event graph without self-loops for which PSA starting in blocking markings may not yield a stationary output

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) $3(\lambda_3), 3(\lambda_3), 2(\lambda_2), 1(2\lambda_1), 3(\lambda_3)$, where a firing with rate $2\lambda_1$ means that transition 1 has enabling degree 2 because there are two tokens in place a. 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.

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 3.

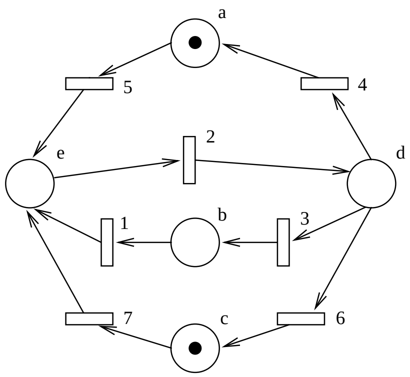

Fig. 3. A free choice net for which PSA starting in blocking markings may not yield a stationary output

The network in Figure 3 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 $(0,\dots,0,2,0,\dots,0)$. 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 packets in place a, while starting with the initial state given in Figure 3, 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.

Actually, a counter example can also be found for a one-bounded free choice net (the previous example is bounded by two) In [4], a one-bounded free choice net is given where all blocking states couple with positive probability in a marking which may be steady state distributed. This example is rather large (20 transitions and 20 places) and the firing sequences involved in the counter example are also long) and is not reported here due to the lack of space.

V. 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 Propp and Wilson Algorithm, this method does not need Markovian assumptions. This technique was used in the past (see for example [12], [7]) to prove the *existence* of stationary regimes for (max,plus) systems in a non constructive way. We will show here how they can also be used to get perfect *samplings* of this stationary regime.

In the following we consider a bounded and live event graph under the following stochastic assumptions on the firing times:
(H_1): $\tau_q(n)$ are i.i.d. and at least one is unbounded: $\exists q \in \mathcal{Q}$ s.t. $\forall x \in \mathbb{R}_+$, $\mathbb{P}(\tau_q > x) > 0$ (for example τ_1 is unbounded).

Actually, even more general conditions (in particular without the unbounded support assumption) are given in [12]. Everything done in the following is also true under these more general technical conditions.

The time evolution of an event graphs can be written under the form of a (max,plus) linear equation of size Q[1]. If $X_i(n)$ is the instant of the end of the nth firing at transition s_i, then there exists a a sequence of (max,plus) matrices $(A(n))_{n\in N}$ such that

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

By definition, $A_{ij}(n) = \bigvee_{c\in\mathcal{C}(i,j)} \sum_{q\in c} \tau_q(n)$, where $\mathcal{C}(i,j)$, the set of all paths from i to j with all places empty except the last one, containing one token. For more on this, see [7], for example.

Under assumptions H_1, $(A(n))_{n\in N}$ 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 $v \in \mathbb{R}^k_{\max}$ is the vector $\gamma(v)$ defined by $\gamma(v)_i = v_i - \min_j v_j$.

Definition 5: A deterministic (max,plus) matrix $D \in \mathbb{R}^{k\times k}_{\max}$ is of rank 1 if all lines are equal up to an additive constant:

$$\forall i,j \quad \exists\, c_{ij}\ s.t.\ c_{ij} + D_{.,i} = D_{.,j}.$$

Alternatively, D is of rank 1 iff

$$\forall i,j,\ \gamma(D_{.i}) = \gamma(D_{.j}) \Leftrightarrow \forall i,j,\ \gamma(D_{i.}) = \gamma(D_{j.}).$$

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 $(A(n))_{n\in N}$ 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 Figure 4.

```
Input A (max,plus) representation $X(n) = X(n-1) \otimes A(n)$ and a sequence of increasing integers $N_1, N_2, \dots$
$m := 1$
repeat
    Compute $B_m := A(-N_m) \otimes \cdots \otimes A(0)$
    $m := m + 1$
until $B_m$ is of rank one
$X(0) := X(-N_m) \otimes B_m$
Output $\gamma(X(0))$
```

Fig. 4. Perfect Simulation Algorithm of (max,plus) linear systems

Under the foregoing assumptions (H_1), it is proved in [12] that 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 conditions $X(0)$). The next theorem shows that the PSA in Figure 4 provides samples with this stationary distribution.

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

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

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

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

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

Proof: For all k, event $E_{k,\varepsilon}$ is defined by

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

where h_{ij} is in the support of $A_{i,j}(n)$ for all $n \in \mathbb{N}$ and 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 the support of $A_{i,j}(n)$ is continuous at h_{ij}. Finally, h_{11} is such that $h_{11} > Q(\max_{(i,j)\neq(1,1)} h_{ij} + \varepsilon)$.

By construction of the deterministic matrix $H = (h_{ij})$, and using the theory of deterministic (max,plus) matrices (see [7], [5]), 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 $k > Q$, $H^k_{ij} = h_{ii_1} + \cdots + h_{i_\ell 1} + (k - \ell - r - 2)h_{11} + h_{1j_1} + \cdots + h_{j_r j}$. Using this form of the matrix H^k, it is straightforward to show that H^k is of rank one, indeed the difference $H^k_{ij} - H^k_{i'j}$ does not depend on j.

Now, if $k > Q$ and ε is small enough, the product $A(1) \otimes \cdots \otimes A(k)$ will also be of rank one, for the same reason:

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

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

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

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

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

From a stationary profile γ_∞, it is possible to get a stationary state of the fork-join network by appending the following steps in the (max,plus) simulation algorithm. The output is a state distributed according to the stationary distribution.

```
Sample of a real non-negative r. v. $d$ independent of everything
$k := 0, M := M_0$ (initial marking)
$X(0) := \gamma(X_\infty)$ (stationary profile)
repeat
    $X(k) := X(k-1) \otimes A(k)$
    $k := k + 1$
until  $X_i(k) > \max_j X_j(0) + d, \ \forall i$
for all 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 $s_i$.
end for
Output $M$
```

VI. 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 [13]. 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 [15] and has been used in [14], 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 $3.819 \ 10^{31}$.

Figure 5 displays the number of iterations for both algorithms, while Figure 6 displays the total

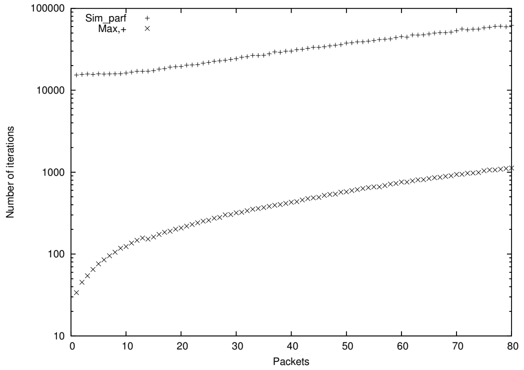

Fig. 5. Average number of iterations for a circuit with $K = 40$ transitions when the number of tokens varies

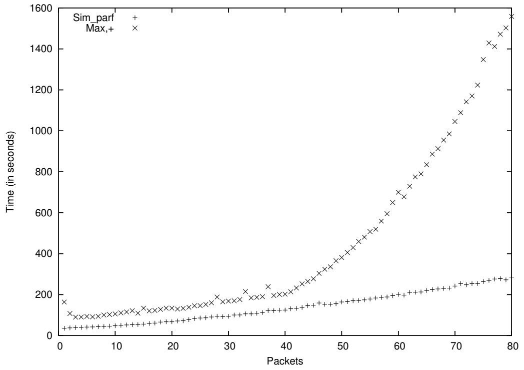

Fig. 6. Average simulation time for a circuit with $K = 40$ transitions when the number of tokens varies

simulation time. Each point is the average of many simulations (enough to guarantee confidence intervals smaller than 1 % of the empirical value).

While the number of iterations before coupling is much smaller for the (max,plus) case, the actual simulation time however is much larger. This is because one step in the (max,plus) algorithm is a product of a large matrix (of size K). One can notice that the time complexity of the (max,plus) algorithm starts to increase rapidly when the number of packets 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 corresponding curves are similar to those in Figures 5 and 6 and are not reported here.

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

VII. Several extensions: heaps of pieces and safe free-choice nets

A. *Heaps of pieces*

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

For $a \in \mathcal{A}$, $C(a)$ is the set of resources used to execute task a. The lower contour $l(a)$ of a is a vector defined on $\mathcal{C}$ giving the relative starting time of the execution of a on $\mathcal{C}$ ($\min_{r \in C(a)} l(a)_r = 0$) with the convention that $l(a)_r = -\infty$ if $r \notin C(a)$. The upper contour of task a is a vector defined on $\mathcal{C}$ giving the relative completion time of the execution of a on $\mathcal{C}$. with the convention that $u(a)_r = -\infty$ if $r \notin C(a)$.

It is shown in [9], that the total execution time of the sequence of tasks $a_0 \dots a_n$ over the resources is given by 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

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

For stochastic heaps, there are two types of randomness. First, for a given a, $l(a)$ and $u(a)$ are sequences of iid random vectors with a given 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 iid Bernoulli distribution (with probability $p_a > 0$).

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

$$(H_2) : \begin{cases} \forall a \in \mathcal{A},\ \{T(a)(n)\}_{n \in \mathbb{N}} \text{ is iid} \\ \forall r \in \mathcal{C},\ \exists a \in \mathcal{A} \mid r \in C(a) \\ \forall r, s \in \mathcal{C}, \exists\ a_1 \dots a_k \in \mathcal{A}, \\ \quad \exists r_0 (=r) \dots r_k (=s) \mid r_{i-1}, r_i \in C(a_i). \end{cases}$$

In words, this assumptions correspond to the fact that all r.v. are iid, all resources are used and that the heap cannot be split into two independent subheaps.

Theorem 10 ([9]): 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.

Theorem 11: Algorithm 7 terminates in finite time w.p. 1 and its output $\gamma(X(0))$ has the stationary distribution of the heap of pieces.

```
Input An infinite Bernoulli sequence in $\mathcal{A}$, $a_1, \ldots, a_n, \ldots$, matrices $T(a_1), \ldots T(a_n), \ldots$ and a sequence of increasing integers $N_1, N_2, \ldots$.
$m := 1$
repeat
    Compute $B_m := T(a_{-N_m}) \otimes \cdots \otimes T(a_0)$
    $m := m + 1$
until $B_m$ is of rank one
$X(0) := X(-N_m) \otimes B_m$
Output $\gamma(X(0))$
```

Fig. 7. Perfect sampling of the stationary distribution of a heap of pieces

Proof: Consider a finite sequence $a_1 \ldots a_n$ 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). Then the sequence $T(a_1) \otimes \cdots \otimes T(a_n) \otimes T(a_n) \otimes \cdots \otimes T(a_1)$ is a matrix of rank one ([9]). Since the probability of each piece is positive and since the selection process is iid, The probability that the sequence $a_1 \ldots a_n a_n \ldots a_1$ occurs is also positive. Using Borel-Cantelli 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 $\gamma(X_0 \otimes B_m) = \gamma(X_\infty \otimes B_m)$ because B_m is of rank 1, and $\gamma(X_\infty \otimes B_m) = \gamma(X_\infty)$ because X_∞ is stationary. ■

B. Safe free choice nets

In this section we consider one-bounded (or safe) free choice nets (SFCNets) with iid stochastic firing times with arbitrary distributions, under *Bernoulli routing* (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 $q \in p^\bullet$). SFCNets can be seen as heaps of pieces (see for example [9]). Here a piece corresponds to a transition and the ressources are the places. The ressources used by transition q, $C(q)$ 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, $C(q) = {}^\bullet q \cup q^\bullet$ and

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

If the firing times are iid in all transitions and if the net is strongly connected, then assumptions H_2 are verified for the corresponding heap of pieces. However, one main difference with free heaps is that 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 corresponding to the pieces with a matrix modelling the reachability graph of the free choice net. This approach has a major drawback: the size of the matrices is $O(P|\mathcal{R}(M_0)|)$ so one loop of the simulation algorithm is in $O(P^3|\mathcal{R}(M_0)|^3)$ which may be too large to be handled by a computer.

Another way, detailed below, is to consider only the initial heap of pieces and at each step, check whether a given new piece can be stacked. The overall complexity of one loop of the simulation algorithm will be in $O(P^3|\mathcal{R}(M_0)|)$.

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 $q_1 \ldots q_n$, the heap has reached the state $X(n) = X(0)T(q_1) \otimes \cdots \otimes T(q_k)$. 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 P-components [6]. For each P-component, when firing transitions $q_1 \ldots q_n$, the latest event within the P-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 P-component. Knowing the position of the tokens in all the P-components covering the SFCNet determines the marking. ■

Here is the main result of this section.

Theorem 13: *i*- Algorithm 8 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 himself that each trajectory simulated by this algorithm 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 *i*, the first step is to show 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.

```
Input $(P, Q, \tau, C)$ a SFCNet with firing times $T = \{\tau_q\}$, Bernoulli routings $\{c_{pq}\}$, a matrix $B := Id$
for all Marking $M \in \mathcal{R}(M_0)$ do
  Compute a heap $X_M(0)$ such that $Mark(X_M(0)) = M$.
end for
repeat
  Pick a transition $q$ w.p. $c_{\bullet q q}/\Gamma$
  for all reachable marking $M$ do
    If $q$ is enabled in $Mark(X_M(k-1))$
    then $B := T(q) \otimes B$ end if
    $X_M(k) := X_M(0) \otimes B$
    $k := k + 1$
  end for
until $B$ is of rank one and $Mark(X_M(k))$ is the same for all markings $M \in \mathcal{R}(M_0)$.
Output $\gamma(X_M(k))$, for one $M \in \mathcal{R}(M_0)$.
```

Fig. 8. PSA of the stationary profile of the heap of a SFCNet

Consider a sequence of pieces that first assure coupling of the marking. Once a common marking is reached for all heaps, one only needs to exhibit a firing sequence such that the corresponding pieces form of matrix of rank one. The construction used in Theorem 11 cannot be used directly here because a sequence $q_1 \dots q_n q_n \dots q_1$ may not be enabled. Instead, one must construct a similar sequence by parts. starting from the common marking, a blocking marking M_q (for an arbitrary transition q) can be reached w.p.p. From M_q there exists a valid firing sequence σ_1 involving transition q_1, leading back to marking M_q. The same can be done for all the transitions in the SFCNet. Combining all sequences enables a firing sequence $\sigma_1 \dots \sigma_Q \sigma_Q \dots \sigma_1$ which can be fired w.p.p. This yields a matrix with rank 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 compute $Mark(X_M(k))$.

Unlike for 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. Note that computing the stationary distribution is already very hard with two or three nodes, so that sampling remains interesting.

Several generalizations are possible.
- If conflicts are solved using the *race policy* (the transition with the smallest firing time wins the conflict), this is also amenable to 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

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

- Iid assumptions for firing times can be replaced by more general stationary and ergodic assumptions. However this introduces several technical difficulties similar to those in [1], which are beyond the scope of this paper

Acknowledgement

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

REFERENCES

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

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

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

[4] A. Bouillard. *Optimisation et analyse probabiliste de systèmes à événements discrets*. PhD thesis, École Normale Supérieure de Lyon, 2005. In French.

[5] A. Bouillard and B. Gaujal. 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, 2001. IFAC.

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

[7] G. J. Olsder F. Baccelli, G. Gohen and J.-P. Quadrat. *Synchronization and Linearity*. Wiley, 1992.

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

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

[10] B. Gaujal, S.Haar, and J. Mairesse. 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, 2003.

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

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

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

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

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