Perfect sampling of Jackson Queueing Networks

Ana Bušić, Stéphane Durand, Bruno Gaujal, Florence Perronnin

To cite this version:

Ana Bušić, Stéphane Durand, Bruno Gaujal, Florence Perronnin. Perfect sampling of Jackson Queueing Networks. [Research Report] RR-8332, INRIA. 2013, pp.32. hal-00851331v2

HAL Id: hal-00851331

https://hal.inria.fr/hal-00851331v2

Submitted on 4 Apr 2014

Perfect Sampling of Jackson Queueing Networks

Ana Bušić , Stéphane Durand , Bruno Gaujal , Florence Perronnin

RESEARCH REPORT
N° 8332
July 2013
Project-Team Mescal

ISSN 0249-6399 ISRN INRIA/RR--8332--FR+ENG

Perfect Sampling of Jackson Queueing Networks

Ana Bušić [*] [†], Stéphane Durand [‡] [*], Bruno Gaujal [*] [§] [¶],
Florence Perronnin [§] [¶] [*]

Project-Team Mescal

Research Report n° 8332 — version 2[‖] — initial version July 2013 —
revised version December 2013 — 32 pages

Abstract: We consider open Jackson networks with losses with mixed finite and infinite queues and analyze the efficiency of sampling from their exact stationary distribution. We show that perfect sampling is possible, although the underlying Markov chain may have an infinite state space. The main idea is to use a Jackson network with infinite buffers (that has a product form stationary distribution) to bound the number of initial conditions to be considered in the coupling from the past scheme. We also provide bounds on the sampling time of this new perfect sampling algorithm for acyclic or hyper-stable networks. These bounds show that the new algorithm is considerably more efficient than existing perfect samplers even in the case where all queues are finite. We illustrate this efficiency through numerical experiments. We also extend our approach to non-monotone networks such as queuing networks with negative customers.

Key-words: Perfect Simulation, Markov Chain, Jackson networks, bounding process

[*] Inria
[†] Département d'Informatique de l'ENS (DI ENS) Paris, France
[‡] ENS of Lyon, France
[§] Univ. Grenoble Alpes, LIG, F-38000 Grenoble, France
[¶] CNRS, LIG, F-38000 Grenoble, France
[‖] Additional material: unstable bounding network (section 3.2), Traffic equation solution for unstable networks (Appendix A), Extended related work and small improvements.

**RESEARCH CENTRE
GRENOBLE – RHÔNE-ALPES**

Inovallée
655 avenue de l'Europe Montbonnot
38334 Saint Ismier Cedex

Simulation parfaite de réseaux de Jackson

Résumé : On considère les réseaux de Jackson avec perte comportant des files finies et infinies, et l'on s'intéresse à l'efficacité des techniques d'échantillonnage de leur distribution stationnaire exacte. Nous démontrons que la simulation parfaite est possible même si la chaîne de Markov sous-jacente a un espace d'états potentiellement infini. L'idée principale est d'utiliser un réseau de Jackson aux files infinies (qui admet une distribution de forme-produit) pour borner les conditions initiales à considérer dans l'algorithme de simulation parfaite. Nous donnons également des bornes sur le temps d'échantillonnage de ce nouvel algorithme dans le cas des réseaux acycliques, ainsi que pour des réseaux hyperstables. Ces bornes prouvent que le nouvel algorithme est considérablement plus efficace que les échantillonneurs parfaits acuels, même dans le cas où toutes les files sont finies. Nous illustrons cette efficacité par des expériences numériques. Enfin, nous généralisons notre approche au cas des réseaux non-monotones comme les réseaux aux clients négatifs.

Mots-clés : Simulation parfaite, réseaux de Jackson, chaînes de Markov, processus bornant

1 Introduction

The stationary behavior of queuing networks can only be obtained quite efficiently under specific assumptions that yield the so called product-form property; e.g., [Kelly, 1979]. This property means that the stationary probability distribution of these networks can be decomposed, up to a normalizing constant, in the product of the marginal distributions of each network node (or queue). In several cases, product-form queuing networks are restrictive because they often assume that nodes have infinite buffer sizes or that the behavior of a network node does not depend on the state of other nodes; e.g., [Jackson, 1963]. In the context of Internet networks, blocking and rejection mechanisms arise due to finite-buffer constraints and state-dependent routing. While it is possible to obtain the stationary distribution of non-product-form queuing networks through the solution of a set of linear equations, i.e., the global-balance equations [Bolch et al., 2005], the huge size of their state space makes this approach of practical interest only for small networks. In this setting, simulation is a useful approach to obtain robust measures and insights on the stationary performance.

At the cost of a slightly higher computational complexity than Monte Carlo simulation, the exact stationary distribution can be sampled in finite time using a technique called coupling from the past (see the seminal work by Propp and Wilson [Propp and Wilson, 1996]). Unlike Monte Carlo simulation, this powerful technique produces independent samples of ergodic finite Markov chains *exactly distributed* according to their stationary distribution. For this reason, this technique is also known as perfect sampling algorithm (PSA) and will be denoted by PSA in the remainder of the paper.

This technique has been used to design simulation algorithms for queuing networks with finite capacity buffers and rather general routing policies [Vincent, 2005]. The main assumption needed to make these algorithms work is that the state space is finite (or equivalently that the buffer capacities are all finite).

Contributions

In this paper we present a new perfect simulation algorithm for queuing networks, that can handle finite and infinite buffers at the same time, with rejection of jobs arriving at a saturated queue (Sec. 3). The proposed sampling algorithm works as follows.

1. First, we sample the stationary behavior of the same network but with infinite buffers:
   - if this infinite-capacity network is stable, we solve the linear traffic equation and sample the steady-state using product-form property.
   - If the bounding network is unstable, we incidentally provide an algorithm (Algorithm 4 in Appendix A) for solving the traffic equations with cubic complexity in the number of queues $O(M^3)$, whereas classical algorithms typically need $O(M^4)$ [Goodman and Massey, 1984]. Then we can sample the asymptotic behavior by setting the state $+\infty$ in the unstable queues and using the product-form property on the stable sub-network.
2. This infinite-capacity network is a bounding process of the original queuing network.
3. We reverse in time the trajectory of the bounding network.
4. With a careful coupling of events (provided in Sec. 2.2), the obtained state can be used as a maximum starting point of the forward trajectory of the original network (up to its capacity) in the perfect sampling algorithm.

We also derive a bound on its sampling time complexity and show that it does not depend on the size of the state space, when the network is acyclic, or under a hyperstability condition (Sec. 4).

We then conduct numerical experiments to validate the approach, compare its efficiency to classical perfect sampling, and assess the tightness of our theoretical complexity bound (Sec. 6). Additionally, we extend our approach (Sec. 7) to networks with negative customers. While the treatment of negative customers in a network has an interest *per se*, this part also shows that the idea to use a dominating reversed chain can be used for non-monotone chains by combining the dominating reversed chain technique with the construction of envelope (or sandwich) processes.

Related work.

As mentioned before, the original perfect sampling algorithm has been adapted for the simulation of monotone queuing networks in [Vincent, 2005]. The complexity of this algorithm has been analyzed in [Dopper et al., 2006] for acyclic networks, while cyclic networks have been studied in [Anselmi and Gaujal, 2011, Anselmi and Gaujal, 2013].

A series of papers propose new perfect sampling algorithms, introducing several ideas such as *envelopes* (or sandwich process), as in [Gaujal et al., 2008, Kendall and Møller, 2000, Pin et al., 2011], *splitting* [Busic et al., 2008] and *skipping* [Busic et al., 2012b] that improve the runtime of perfect sampling for non-monotone queuing networks. However, in all cases, it is essential that the underlying Markov chain has a finite state space.

In [Kendall, 2005], the concept of dominating process is introduced, associated with coupling from the past (DomCFTP). When a monotone Markov chain (finite or infinite) admits a dominating chain whose stationary distribution can be sampled easily, then one can first sample the dominating chain, then simulate it backward in time, using this reversed dominating chain as the starting point for the forward simulation of the original chain.

Our paper is based on this idea. We show that this construction is possible for sampling the stationary distribution of Jackson networks with finite and infinite queues. The main difficulty of this construction is to find a coupling between the original chain and the reversed dominating chain. This difficulty comes from the following fact. To simulate any Markov chain X_n, one needs to construct a deterministic function ϕ such that $X_n = \phi(X_{n-1}, u_n)$, where u_n forms an iid sequence of uniform random variables. Coupling the reversed dominating chain with the original one corresponds to finding a function ϕ for the original chain and a function β for the reversed dominating chain such that $\phi \circ \beta(x, u) = x$. But this implies that functions ϕ and β are invertible on their first coordinate, for all random u. Should such a coupling be possible, it would prevent the original chain to ever reach coalescence (iterating a bijective ϕ will never decrease the number of trajectories).

Here, we show that in spite of the impossibility to couple the direct original chain and the reversed dominating chain, the construction proposed by Kendall can still be made effective for the particular case of Jackson networks. This is done by modifying the value on u according to the value of the dominating chain. Such a modification can introduce a simulation biais in general, but we prove that this is not the case here.

This paper is an extended version of a conference paper [Busic et al., 2012a] where the construction of the coupled backward bounding process for queuing networks was introduced. While the *existence* of the coupling between the forward and the backward processes was proved in our preliminary work [Busic et al., 2012a], here, we explicitely construct the coupling between the two backward processes $Y(-t)$ and $Y^{\infty}(-t)$ as well as the coupling between the backward and the forward processes, $Y^{\infty}(-t)$ and $X^{(1)}(t)$. As a result, our main algorithm is now effective and ready for straightforward implementation. This explicit construction is also used in the

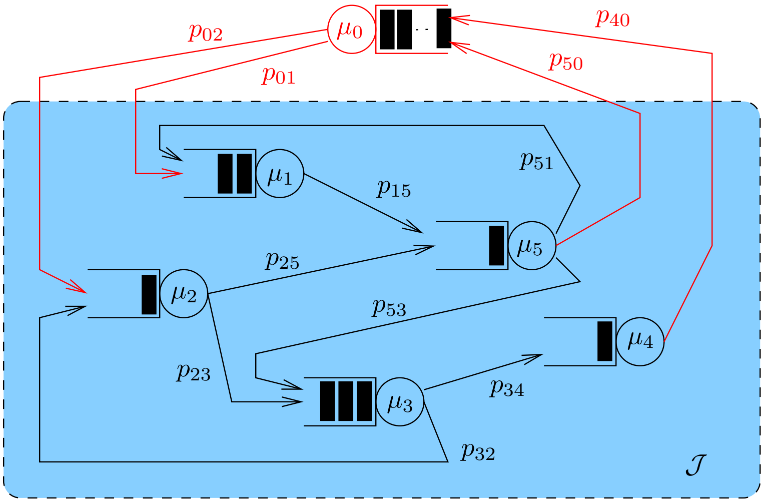


Figure 1: An open Jackson network whose input/output flows are seen as inputs/outputs from/to an additional queue with an infinite backlog.

proofs of Theorems 1 and 2. In this paper, we also provide a derivation of the complexity of the new Perfect Sampling Algorithm based on a Bounding Process (PSA-BP) in two cases: for stable acyclic networks and hyper-stable networks, given in Theorems 2 and 3 that was not present in [Busic et al., 2012a]. We also present new experiments in Section 6, showing the effect of the load in the performance of the prefect sampling algorithm as well as the tightness of the theoretical complexity bounds. Finally, the treatment of negative customers (that make the chain non-monotone), is also new.

2 Queuing network model

We consider an open Jackson queuing network (JQN) $\mathcal{J}$ with M queues. The vector $\mathbf{C} = (C_1, \ldots, C_M)$ denotes the buffer size of each queue. Note that for any i, $C_i \in \mathbb{N} \cup \infty$, *i.e.* the buffer capacities can either be finite or infinite. When a queue is full, all arriving jobs at this queue are lost (no blocking). For $1 \leqslant i \leqslant M$, we denote by $\mathbf{e}_i$ the vector in $\mathbb{Z}^M$ with all the components equal to 0, except for component i that is equal to 1.

To unify the notations, in particular with respect to the exogenous/ endogenous flows of packets, we use a modeling trick: The outside world is seen as one additional queue (numbered 0) with an infinite capacity $C_0 = \infty$, containing an infinite number of jobs initially: $X_0(0) = +\infty$.

Therefore, the input stream in the network with rate λ_0 can be seen as coming from queue 0 whose service rate is $\mu_0 \overset{\text{def}}{=} \lambda_0$. Figure 1 illustrates this construction.

The probability that a job joins queue i, upon arrival to the network, is p_{0i}. In queue i ($i > 0$), each job requires some processing for an exponentially distributed amount of time with mean service rate μ_i. The service discipline of each queue i is work-conserving. Upon completion of service at queue i, a job is sent to queue j with probability p_{ij}, and it is accepted if queue j has an available slot (i.e., if it is non-saturated), otherwise the job is lost. The probability that a job leaves the network after service at i is p_{i0}. We make the assumption that the matrix $P = (p_{ij})_{i,j \in \{0,\ldots,M\}}$ is irreducible. With the trick about Queue 0 representing the outside world, this is a compact way to state the classical assumptions that all queues always get new jobs and that all jobs eventually leave the network.

Under the foregoing assumptions, the vector of the queue occupancies forms a continuous-time Markov chain. The state space is $\mathcal{S} \overset{\text{def}}{=} \{\mathbf{x} \in \mathbb{Z}^M : 0 \leq x_i \leq C_i, \forall i\}$. The notation $X(t) = (X_1(t), \ldots, X_M(t))$ refers to the corresponding uniformized Markov chain in discrete time ($t \in \mathbb{N}$).

Our main notation is summarized in Table 1.

2.1 Discrete-event definition of JQN

The JQN $\mathcal{J}$ with M queues described above can be seen as a discrete-event system with a single type of events, namely $\{r_{ij},\ i,j \in \{0,1,\ldots,M\}\}$ corresponding to the service of one job in queue i that then joins queue j. The dummy queue 0 corresponds to the outside world: an event of type r_{0j} is an exogenous arrival in queue j and an event of type r_{i0} corresponds to the departure of a job from queue i. If queue i is empty then event r_{ij} has no effect on the system. Similarly, if queue j is full (i.e. if $X_j(t) = C_j$) then event r_{ij} only affects queue i. The set of all events is denoted by $\mathcal{R}$.

The rate of event r_{ij} is γ_{ij} and it is independent of M and $\mathbf{C}$, for any i,j. Using the previous description of a JQN, for all $i,j \in \{0,1,\ldots,M\}$, $\gamma_{ij} = \mu_i p_{ij}$. The total event rate $\Gamma \overset{\text{def}}{=} \sum_{i,j=0}^{M} \gamma_{ij}$ is finite (we set arbitrarily $\gamma_{00} = 0$).

The continuous-time Markov chain described above can be transformed into a discrete-time Markov chain $\{X(t)\}_{t\in\mathbb{N}}$ with the same stationary distribution using uniformization by constant Γ. Using the foregoing assumption that the routing matrix P is irreducible, this discrete chain is irreducible and aperiodic. The evolution of the Markov chain $X(t)$ can be written under the form $X(t+1) = \phi(X(t), r_{t+1})$ where r_{t+1} is the $t+1$-st event, distributed according to probabilities $a_{ij} = \gamma_{ij}/\Gamma$, $1 \leqslant i,j \leqslant M$.

The state changes according to events that correspond to the routing of one packet from queue i to queue j, as defined previously.

The forward transition function of the chain $\phi : \mathcal{S} \times \mathcal{R} \to \mathcal{S}$ is defined as follows:

$$\phi(\mathbf{x}, r_{ij}) = \mathbf{x} + (\mathbf{e}_j \mathbb{1}_{\{x_j < C_j\}} - \mathbf{e}_i)\mathbb{1}_{x_i > 0}$$

where $\mathbf{e}_i$ is the unit vector in direction i ($\mathbf{e}_0$ is the zero vector) and $\mathbb{1}_{\mathcal{P}}$ equals 1 if proposition $\mathcal{P}$ is true and 0 otherwise.

This transition function can be extended to any finite sequence $(r_1, r_2, \ldots, r_k) \in \mathcal{R}^k,\ k \in \mathbb{N}$, by defining $\phi : \mathcal{S} \times [0,1]^k \to \mathcal{S}$ recursively:

$$\phi(\mathbf{x}, r_1, r_2, \ldots, r_k) \overset{\text{def}}{=} \phi(\phi(\mathbf{x}, r_1), r_2, \ldots, r_k).$$

2.2 Network with infinite buffers

Starting from $\mathcal{J}$, we construct a new network $\mathcal{J}^\infty$ that is identical to $\mathcal{J}$ except for the buffer sizes: in $\mathcal{J}^\infty$ all queues have infinite capacities. The state space of this new network will be denoted by $\mathcal{S}^\infty \overset{\text{def}}{=} \{\mathbf{x} \in \mathbb{Z}^M : x_i \geq 0, \forall i\}$.

For clarity, we first assume in this section that each queue of the *infinite* JQN is stable, i.e., the total arrival rate at queue i λ_i, determined the traffic equation $\lambda_j = \sum_{i=0}^{M} \lambda_i p_{ij}$ for $1 \leqslant i \leqslant M$, satisfies $\lambda_i < \mu_i$ for $1 \leqslant i \leqslant M$. We will relax this stability assumption in section 3.2.

The corresponding discrete-time Markov chain is denoted $\{X^\infty(t)\}_{t\in\mathbb{N}}$. The evolution of the Markov chain $X^\infty(t)$ can be written in discrete time using the same events as for the finite case,

Glossary

JQN	Jackson queuing network
PSA	Perfect sampling algorithm [Propp and Wilson, 1996]
PSA-BP	Perfect sampling with bounding process (proposed algorithm)

Mathematical conventions

$\mathbf{e}_i$	Unit vector: with zero coordinates except the i-th one that is equal to 1.
$\mathbb{1}_A$	Indicator function: 1 if proposition A is true, 0 otherwise
$x \wedge y$	componentwise minimum between vectors x and y
$x \vee y$	componentwise maximum between vectors x and y
$\preceq$	componentwise partial order

System parameters

M	Number of queues
C_i	Capacity of queue i ($C_i \in \mathbb{N} \cup \{+\infty\}$)
$\mathbf{C}$	Vector of queue capacities
λ_0	Total exogenous job arrival rate (also $\mu_0 = \lambda_0$).
p_{0i}	Prob. that exogenous jobs arrive in queue i
p_{ij}	Routing probability from queue i to j
p_{i0}	Prob. that after service completion at queue i a job leaves the system
μ_i	Service rate of queue i

Algorithm and proof notation

Original network	*Reversed network*	
$\mathcal{J}$	$\mathcal{K}$	Original/reversed Jackson Network (finite and infinite queues)
$\mathcal{J}^\infty$	$\mathcal{K}^\infty$	Bounding infinite-queue JQN (resp. reversed)
p_{ij}	q_{ij}	Routing prob. from queue i to j in $\mathcal{J}$ (resp. $\mathcal{K}$). $q_{ij} = p_{ji}\lambda_j/\lambda_i$
λ_i		Total arrival rate in queue i ($\lambda_i = \sum_{j=0}^{M} \lambda_j p_{ji}$).
$\mathcal{R}$		Set of all possible events
r_{ij}		Event "job leaving queue i and joining queue j"
γ_{ij}		Rate of event r_{ij} ($\gamma_{ij} = \mu_i p_{ij}$)
Γ		Total event rate ($\Gamma \overset{\text{def}}{=} \sum_{ij} \gamma_{ij}$)
a_{ij}	b_{ij}	Prob. of event r_{ij} in the original/reversed chain: $a_{ij} = \dfrac{\gamma_{ij}}{\Gamma}$, $b_{ij} = q_{ij}\dfrac{\mu_i}{\Gamma}$.
A_{ij}	B_{ij}	Interval of [0,1] with length a_{ij} (resp. b_{ij})
v_t	w_t	Coupled sequences of events for the original / reversed chains
$\mathcal{S}$		State space of the original and reversed CTMC
$\mathcal{S}^\infty$		State space of the infinite-queue original and reversed CTMC
$X(t)$	$Y(t)$	Uniformized, discrete-time (reversed) Markov chain
$X^\infty(t)$	$Y^\infty(t)$	Uniformized bounding DTMC for $\mathcal{J}^\infty$ (resp. $\mathcal{K}^\infty$)
$X_t^{(1)}(\cdot)$		Upper trajectory starting at time $-t$
$X_t^{(2)}(\cdot)$		Lower trajectory starting at time $-t$
$\phi(\mathbf{x}, r)$	$\beta(\mathbf{x}, r)$	Transition function for $\mathcal{J}$ (resp. $\mathcal{K}$) in state $\mathbf{x}$ under event r
$\phi^\infty(\mathbf{x}, r)$	$\beta^\infty(\mathbf{x}, r)$	Transition function for $\mathcal{J}^\infty$ (resp. $\mathcal{K}^\infty$)

Table 1: Main notation and glossary

under the form $X^\infty(t+1) = \phi^\infty(X^\infty(t), r_{t+1})$ where the function ϕ^∞ is defined by:

$$\phi^\infty(\mathbf{x}, r_{ij}) = \mathbf{x} + (\mathbf{e}_j - \mathbf{e}_i)\mathbb{1}_{x_i>0}$$

The functions ϕ^∞ and ϕ are related in the following way.

$$\phi(x, r) = \phi^\infty(x, r) \wedge \mathbf{C} \quad \text{for all event } r. \tag{1}$$

where the notation $\mathbf{x} \wedge \mathbf{y}$ (resp. $\mathbf{x} \vee \mathbf{y}$) denotes the componentwise minimum (resp. maximum) of $\mathbf{x}$ and $\mathbf{y}$.

As before, $\phi^\infty : \mathcal{S}^\infty \times \mathcal{R}^k \to \mathcal{S}^\infty$ is defined recursively:

$$\phi^\infty(\mathbf{x}, r_1, r_2, \dots, r_k) \overset{\text{def}}{=} \phi^\infty(\phi^\infty(\mathbf{x}, r_1), r_2, \dots, r_k).$$

Let us we consider the usual product partial order of states: for $\mathbf{x}, \mathbf{y} \in \mathcal{S}^\infty$,

$$\mathbf{x} \preceq \mathbf{y} \quad \text{iff} \quad x_i \leqslant y_i,\ 1 \leqslant i \leqslant M.$$

Using this order, the new chain is a *bounding process*[1] of the original chain in the following sense:

If $X(0) \preceq X^\infty(0)$, then for all k and all sequence of events $r_1, \dots, r_k$, then $X(k) = \phi(X(0), r_1, r_2, \dots, r_k) \preceq \phi^\infty(X(0), r_1, r_2, \dots, r_k) = X^\infty(k)$.

This results from the following properties:

Lemma 1. *For any $r \in \mathcal{R}$, and any $\mathbf{x} \in \mathcal{S}$, functions ϕ and ϕ^∞ satisfy:*

$$\phi(\mathbf{x}, r) \preceq \phi^\infty(\mathbf{x}, r).$$

Proof. This follows directly from the fact that for any r, and any $\mathbf{x} \in \mathcal{S}$, $\phi(\mathbf{x}, r) = \phi^\infty(\mathbf{x}, r) \wedge \mathbf{C}$. □ □

To establish the comparison between the two chains, we also use the fact that ϕ^∞ is monotone in $\mathbf{x}$. We show similarly that the original system ϕ is also monotone, for later use.

Lemma 2. *(monotony) For any $r \in \mathcal{R}$, and any $\mathbf{x}, \mathbf{y} \in \mathcal{S}^\infty$,*

$$\mathbf{x} \preceq \mathbf{y} \ \Rightarrow\ \phi^\infty(\mathbf{x}, r) \preceq \phi^\infty(\mathbf{y}, r) \ \text{ and } \ \phi(\mathbf{x}, r) \preceq \phi(\mathbf{y}, r).$$

Proof. Let $r = r_{ij} \in \mathcal{R}$ and $\mathbf{x} \preceq \mathbf{y}$. Clearly, if $x_i > 0$ or if $y_i = 0$ then $\mathbb{1}_{x_i>0} = \mathbb{1}_{y_i>0}$, in which case we have $\phi^\infty(\mathbf{x}, r_{ij}) - \phi^\infty(\mathbf{y}, r_{ij}) = \mathbf{y} - \mathbf{x} \succeq 0$. Now if $x_i = 0$ and $y_i > 0$ then :

$$\phi^\infty(\mathbf{y}, r) - \phi^\infty(\mathbf{x}, r) = \mathbf{y} - \mathbf{x} + \mathbf{e}_j - \mathbf{e}_i$$

To prove positivity we only need to check the i-th coordinate: $y_i - x_i - 1 \geqslant 0$ by hypothesis. So ϕ^∞ is monotonic.

From (1), ϕ is the minimum of a monotonic function and a constant, so it is monotonic too. □ □

Combining the two lemmas, we get the sample path comparison of the two systems:

Proposition 1. *For any $(r_{-t+1}, \dots, r_0) \in [0,1]^t$, and any $\mathbf{x} \in \mathcal{S}$, $\mathbf{y} \in \mathcal{S}^\infty$,*

$$\mathbf{x} \preceq \mathbf{y} \ \Rightarrow\ \phi(\mathbf{x}, r_{-t+1}, \dots, r_0) \preceq \phi^\infty(\mathbf{y}, r_{-t+1}, \dots, r_0).$$

[1] Bounding processes are called dominating processes in [Kendall, 2005]

Proof. By induction on t. For $t = 1$, and any $\mathbf{x} \preceq \mathbf{y}$, using first Lemma 1 and then Lemma 2, we have $\phi(\mathbf{x}, r_0) \preceq \phi^\infty(\mathbf{x}, r_0) \preceq \phi^\infty(\mathbf{y}, r_0)$.

Assume now the induction statement is valid for $t-1$. Let $\mathbf{x} \preceq \mathbf{y}$, and denote $\mathbf{x}' = \phi(\mathbf{x}, r_{-t+1})$, and $\mathbf{y}' = \phi^\infty(\mathbf{y}, r_{-t+1})$. Using Lemmas 1 and 2, $\mathbf{x}' = \phi(\mathbf{x}, r_{-t+1}) \preceq \phi^\infty(\mathbf{x}, r_{-t+1}) \preceq \phi^\infty(\mathbf{y}, r_{-t+1}) = \mathbf{y}'$. Now $\phi(\mathbf{x}, r_{-t+1}, \dots, r_0) = \phi(\mathbf{x}', r_{-t+2}, \dots, r_0)$ and $\phi^\infty(\mathbf{y}, r_{-t+1}, \dots, r_0) = \phi^\infty(\mathbf{y}', r_{-t+2}, \dots, r_0)$, so

$$\phi(\mathbf{x}, r_{-t+1}, \dots, r_0) \preceq \phi^\infty(\mathbf{y}', r_{-t+2}, \dots, r_0)$$

by induction hypothesis. □ □

The new Markov chain $\{X^\infty(t)\}_{t\in\mathbb{N}}$ has three interesting properties:

1. A consequence of Proposition 1 is that the process $\{X^\infty(t)\}_{t\in\mathbb{N}}$ is a bounding process of the original process $\{X(t)\}_{t\in\mathbb{N}}$;

2. Under the stability assumption, $\{X^\infty(t)\}_{t\in\mathbb{N}}$ has a unique stationary distribution with the product form property:

$$\pi^\infty(x_1, \dots, x_M) = \prod_{i=1}^{M} (1-\rho_i)\rho_i^{x_i}, \tag{2}$$

   where $\rho_i \overset{\text{def}}{=} \frac{\lambda_i}{\mu_i}$ is the ratio of the total arrival rate in queue i, λ_i over the service rate μ_i. The arrival rate in queue i satisfies the routing balance equations:

$$\lambda_i = \sum_{j=0}^{M} p_{ji}\lambda_j, \text{ for all } i. \tag{3}$$

3. The time-reversed chain $\{Y^\infty(t)\}_{t\in\mathbb{N}}$ of the Markov chain $\{X^\infty(t)\}_{t\in\mathbb{N}}$ can be easily constructed.

Reversed chain.

The reversed chain $\{Y^\infty(t)\}_{t\in\mathbb{N}}$ can also be modeled as a Jackson network $\mathcal{K}^\infty$ with M queues with service rate μ_i in queue i, whose routing probabilities $q_{ij} \overset{\text{def}}{=} p_{ji}.\lambda_j/\lambda_i$, the probability that a customer leaves the system at queue i is $q_{i0} \overset{\text{def}}{=} p_{0i}.\lambda_0/\lambda_i$, and the exogenous arrival rate in queue j is $p_{j0}\lambda_j$ [Chen and Yao, 2001, Th. 2.11, p. 31].

Therefore, the set of events that modify the state are the same as in the direct chain. It is the set of routing events from one queue to another. However these events will not have the same probability as for the direct chain. For example in the Jackson network given in Figure 1, the event $r_{2,5}$ has a positive probability for the direct chain and a probability zero for the reversed chain. The probability of event r_{ij} in the reversed chain is denoted by $b_{ij} = q_{ij}\mu_i/\Gamma$.

Similarly to the forward construction of the Markov chain, the backward transition function $\beta^\infty : \mathcal{S}^\infty \times \mathcal{R} \to \mathcal{S}^\infty$ is defined as follows:

$$\beta^\infty(\mathbf{y}, r_{ij}) = \mathbf{y} + (\mathbf{e}_j - \mathbf{e}_i)\mathbb{1}_{y_i>0} \quad \text{for all } i,j.$$

Remark 1. *β^∞ has the same mathematical expression as function ϕ^∞, because both transition functions are defined over* events *rather than random innovations. The dynamics of the forward and backward Jackson networks differ through the event probabilities.*

Coupling the direct and the reversed chains.

The reversed chain can be coupled with the forward chain using an event correspondence. To guarantee the fact that events will be drawn with the correct probabilities in the reversed as well as in the direct way, this coupling will depend on the trajectory of a stationary reversed chain.

Let $Y^\infty(0)$ be a sample of the infinite reversed chain, generated according to its product form stationary distribution π^∞. Given an infinite sequence of i.i.d. uniformly distributed random variables in $[0,1]$ $U_{-1}, U_{-2}, \cdots$, we construct an infinite sequence of events $w_{-1}, w_{-2}, \cdots$ for the reversed chain as follows:

The probability of event r_{ij} in the reversed Jackson network $\mathcal{K}^\infty$ is b_{ij}. The corresponding intervals B_{ij} (of size b_{ij}) are ordered to cover $[0,1]$ using the alpha-numerical order (see Fig. 2):

$$B_{0,1}, B_{0,2}, \ldots B_{0,M}, B_{1,0}, B_{1,1}, \ldots, \ldots, B_{M,M}.$$

Now, if $U_{-t} \in B_{ij}$, then $w_{-t} = r_{ij}$.

At this point, we can construct a trajectory of the reversed chain starting from $Y^\infty(0)$ and using this sequence of events: for all $t > 0$, $Y^\infty(-t) = \beta^\infty(Y^\infty(-t+1), w_{-t})$.

As for the direct chain, the coupled event sequence $v_{-1}, \ldots, v_{-t}, \ldots$ is built as follows:

For the direct chain, the event r_{ij} has probability a_{ij}. We also order the intervals A_{ij} of size a_{ij} in alpha-numerical order:

$$A_{0,1}, A_{0,2}, \ldots A_{0,M}, A_{1,0}, A_{1,1}, \ldots, \ldots, A_{M,M}.$$

- If $Y^\infty(-t) \neq Y^\infty(-t+1)$ then v_{-t} is set to $v_{-t} = (w_{-t})^{-1}$ (the inverse of routing event r_{ij} is the routing in the other direction: $(r_{ij})^{-1} = r_{j,i}$).
- If $Y^\infty(-t) = Y^\infty(-t+1)$ then w_{-t} must be of the form $w_{-t} = r_{i_0 j}$, where queue i_0 is empty under the state $Y^\infty(-t+1)$. This means that the random variable U_{-t} has fallen in the interval $B_{i_0,j}$.

  The traffic equation (3) for the Markov chain implies that

  $$\forall i, \sum_j a_{ij} = \sum_j b_{ij}. \tag{4}$$

  This means that the corresponding intervals in $[0,1]$ coincide for all i: $\cup_j A_{ij} = \cup_j B_{ij}$, so that U_{-t} also belongs to an interval of type $A_{i_0,k}$ for the same empty queue i_0 and some k. Then, the direct event v_{-t} is set to $r_{i_0 k}$.

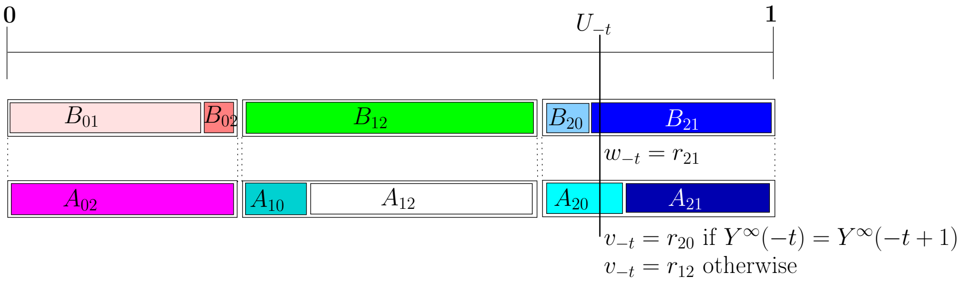


Figure 2: Event probability intervals of the reversed and direct networks for a 3-queue example network.

This coupling between the direct and the reversed chain ensures the following properties:

Property 1. *The trajectory of the reversed chain from time* 0 *to time* $-t$*, starting in* $Y^\infty(0)$ *and the trajectory of the direct chain from time* $-t$ *to time* 0*, starting in* $Y^\infty(-t)$*, coincide:*

$$\forall t' \in \{0, 1, \dots, t\} \quad X^\infty(-t') = \phi^\infty(Y^\infty(-t), v_{-t}, \dots v_{-t'+1}) = Y^\infty(-t'). \tag{5}$$

Property 2. *For any* t*, the probability* $\mathbb{P}(v_{-t} = r_{ij}) = a_{ij}$ *and* v_{-t+1} *is independent of* v_{-t}*.*

Indeed,

$$\begin{aligned}
\mathbb{P}(v_{-t} = r_{ij} | v_{-t-1}) &= \mathbb{P}(v_{-t} = r_{ij} | v_{-t-1} \wedge Y^\infty(-t) = Y^\infty(-t+1)) \\
&\quad + \mathbb{P}(v_{-t} = r_{ij} | v_{-t-1} \wedge Y^\infty(-t) \neq Y^\infty(-t+1)) \\
&= \mathbb{P}(w_{-t} = r_{ji} | v_{-t-1} \wedge Y^\infty(-t) \neq Y^\infty(-t+1)) \\
&\quad + \mathbb{P}(v_{-t} = r_{ij} | v_{-t-1} \wedge Y^\infty(-t) = Y^\infty(-t+1)) \\
&= b_{ji}\rho_j + a_{ij}(1 - \rho_i) \\
&= (\mu_j q_{ji} \rho_j + \mu_i p_{ij}(1 - \rho_i))/\Gamma \\
&= (\mu_j p_{ij} \frac{\lambda_i}{\lambda_j} \rho_j + \mu_i p_{ij}(1 - \rho_i))/\Gamma \\
&= \mu_i p_{ij}/\Gamma \\
&= a_{ij}.
\end{aligned}$$

Actually, it is even possible to simplify the coupling between the two chains because if queue i_0 is empty, it does not matter which second queue is chosen for the routing event: both $r_{i_0 j}$ and $r_{i_0 k}$ have a null effect on the state.

Proposition 2. *Here is a simplified coupling that generates the same trajectories as the previous coupling:*
If $Y^\infty(-t) \neq Y^\infty(-t+1)$ *then* $v_{-t} = (w_{-t})^{-1}$
If $Y^\infty(-t) = Y^\infty(-t+1)$ *then* $v_{-t} = w_{-t}$*.*

3 Perfect Sampling Algorithm with a Bounding Process

We first show Kendall's idea can be made effective with the above coupling to achieve efficient perfect simulation when the JQN $\mathcal{J}$ is is bounded by a stable infinite-queue JQN $\mathcal{J}^\infty$. Then in section 3.2 we relax this assumption and provide an effective algorithm for any stable JQN.

3.1 When the infinite-queue network $\mathcal{J}^\infty$ is stable

The chain $Y^\infty(-t)$ can be used to obtain a bound for the initial condition of the original chain $X(t)$, as shown in Algorithm 1. Let us describe how this algorithm works. First, generate at time 0 a stationary sample $Y^\infty(0)$ of the infinite system according to the product form distribution π^∞.

Starting from $Y^\infty(0)$, construct a backward trajectory up to time $-t$, using the β^∞ function described above and the sequence of events $(w_{-u})_{u\in\mathbb{N}}$: $Y^\infty(-u) = \beta^\infty(Y^\infty(-u+1), w_{-t})$.

At time $-t$, start two forward simulations of the original network $\mathcal{J}$ using the sequence of events v_{-t}, constructed using the previous coupling given in Proposition 2, and starting from states $X^{(2)}(-t) = 0$ and $X^{(1)}(-t) = Y^\infty(-t) \wedge \mathbf{C}$. If the trajectories do not coalesce at time 0 (*i.e.* $X^{(2)}(0)$ and $X^{(1)}(0)$ are not equal), then double the length of the simulation time to $2t$. As shown in the proof below, this does not introduce a bias in the output sample.

Algorithm 1: PSA with a Bounding Process (PSA-BP)

```
Data:
(1) An infinite i.i.d. sequence $\{U_{-n}\}_{n\in\mathbb{N}}$ with $U_{-n}$ uniformly distributed in $[0,1]$ with the
associated sequence of events $\{w_{-n}\}_{n\in\mathbb{N}} : w_{-n} = r_{ij}$ if $U_{-n} \in B_{ij}$.
(2) A state $Y^\infty$ generated according to the distribution $\pi^\infty$.
Result: A state sampled from the stationary distribution of $\mathcal{J}$
begin
    $t := 1$;
    repeat
        for $k = \lfloor t/2 \rfloor$ to $t-1$ do
            $Y^\infty := \beta^\infty(Y^\infty, w_{-k})$;
            $v_{-k} = \begin{cases} (w_{-k})^{-1} & \text{if } Y^\infty \text{ has changed} \\ w_{-k} & \text{otherwise.} \end{cases}$
        $X^{(1)} := Y^\infty \wedge \mathbf{C}$; $X^{(2)} := 0$;
        for $i = t-1$ downto $0$ do
            $X^{(1)} := \phi(X^{(1)}, v_{-i})$;
            $X^{(2)} := \phi(X^{(2)}, v_{-i})$;
        $t := 2t$;
    until $X^{(1)} = X^{(2)}$ ;
    return $X^{(1)}$;
end
```

Theorem 1. *The Algorithm PSA-BP terminates with probability* 1. *The output of Algorithm PSA-BP is a state whose distribution is the stationary distribution of network $\mathcal{J}$.*

Proof. We first prove that the Algorithm PSA-BP terminates with probability 1. The random variable $Y^\infty(-t)$ is distributed according to the stationary distribution π^∞. Thus, the stability assumption for chain $\{Y^\infty(u)\}_{u\in\mathbb{N}}$ implies that we have $\mathbb{P}(Y^\infty(-t) = \mathbf{0}) > 0$ (the zero state is to be understood componentwise). In that case, since $X^{(1)}(\cdot)$ and $X^{(2)}(\cdot)$ are both bounded by $\{Y^\infty(u)\}_{u\in\mathbb{N}}$, we have that $X^{(1)}(-t) = X^{(2)}(-t)$, so we also have $X^{(1)}(0) = X^{(2)}(0)$ and the algorithm terminates. By the Borel-Cantelli lemma, this happens almost surely in finite time so Algorithm PSA-BP terminates with probability 1.

Now, let us show that the output of Algorithm PSA-BP is a state whose distribution is the stationary distribution of network $\mathcal{J}$.

Let us consider the Markov chain $\{Y^\infty(u)\}_{u\in\mathbb{N}}$, the reversed chain of $\{X^\infty(u)\}_{u\in\mathbb{N}}$, and $\{Y(u)\}_{u\in\mathbb{N}}$, the reversed chain of $\{X(u)\}_{u\in\mathbb{N}}$, with initial conditions $Y^\infty(0)$ and $Y(0)$ respectively, distributed according to the stationary distributions π^∞ (resp. π).

Since $X^\infty(\cdot)$ is a bounding process of $X(\cdot)$, the reversed process $Y^\infty(\cdot)$ is also a bounding process of the reversed process $Y(\cdot)$. Now Strassen's theorem and the Markovian property of the reversed processes imply that there exists:

- A joint distribution ν of $Y^\infty(0)$ and $Y(0)$, with marginals π^∞ and π, and such that $Y(0) \preceq Y^\infty(0)$ a.s.,
- and a backward transition function $\beta : \mathcal{S} \times \mathcal{R} \to \mathcal{S}$ for the chain $Y(\cdot)$ is defined by:

$$\beta(\mathbf{y}, r_{ij}) = \mathbf{y} + (\mathbf{e}_j - \mathbf{e}_i)\mathbb{1}_{y_i>0} \wedge \mathbf{C}.$$

The backward function is such that for any $\mathbf{x} \preceq \mathbf{y}$ and any $(w_{-1}, \ldots, w_{-t}) \in \mathcal{R}^t$,

$$\beta(\mathbf{x}, w_{-1}, w_{-2}, \ldots, w_{-t}) \preceq \beta^{\infty}(\mathbf{y}, w_{-1}, w_{-2}, \ldots, w_{-t}). \tag{6}$$

In the proposed PSA-BP Algorithm, the variable $Y^{\infty}(0)$ is generated according to the stationary distribution of $X^{\infty}(\cdot)$. We define $Y(0)$ such that $(Y(0), Y^{\infty}(0))$ are distributed according to ν. We do not need to actually construct $Y(0)$. We only need to assess its existence. By definition, $Y(0)$ has the stationary distribution of $\mathcal{K}$. Now, for any deterministic time t, let us define $Y(-t) = \beta(Y(0), w_{-1}, w_{-2}, \ldots, w_{-t})$, using the backward transition function β of the chain $Y(t)$ (see the blue dotted line in Figure 3), so that it also has the stationary distribution of $\mathcal{K}$. By definition of β, from relation (6) one has $Y(-t) \preceq Y^{\infty}(-t)$ for all t.

Now, let us define the direct processes $\{X^{(1)}(-t;u)\}_{u \geqslant -t}$, $\{X(-t;u)\}_{u \geqslant -t}$ and $\{X^{(2)}(-t;u)\}_{u \geqslant -t}$. The three processes start at time $-t$ with respective initial values

$$\begin{aligned} X^{(1)}(-t;-t) &= Y^{\infty}(-t) \wedge \mathbf{C}, \\ X(-t;-t) &= Y(-t), \\ X^{(2)}(-t;-t) &= 0. \end{aligned}$$

For any time $-u$ such that $-t \leqslant -u \leqslant 0$,

$$\begin{aligned} X^{(1)}(-t;-u) &= \phi(X^{(1)}(-t;-t), v_{-t}, \ldots, v_{-u+1}); \\ X(-t;-u) &= \phi(X(-t;-t), v_{-t}, \ldots, v_{-u+1}); \\ X^{(2)}(-t;-u) &= \phi(X^{(2)}(-t;-t), v_{-t}, \ldots, v_{-u+1}); \end{aligned}$$

where the sequence of events $(v_{-t}, \ldots, v_{-u+1})$ is defined as in Proposition 2.

By construction, $X(-t;-u)$ has the stationary distribution of $\mathcal{J}$, for any t and any $u \leqslant t$.

Also, at the starting time in the past (*i.e.* at time $-t$), one has

$$X^{(1)}(-t;-t) \succeq X(-t;-t) \succeq X^{(2)}(-t;-t).$$

Moving forward in time, the monotony of the chain implies that at time 0,

$$X^{(1)}(-t;0) \succeq X(-t;0) \succeq X^{(2)}(-t;0).$$

Therefore, if $X^{(1)}(-t;0)$ and $X^{(2)}(-t;0)$ coalesce, they are also equal to $X(-t;0)$. This construction is illustrated in Figure 3.

Now, let us show that this common value does not depend on t: We will prove that if $X^{(1)}(-t;0) = X^{(2)}(-t;0) = X(-t;0)$, then for any $s \geqslant t$, then

$$X^{(1)}(-s;0) = X^{(2)}(-s;0) = X(-s;0) = X(-t;0).$$

On one hand, $X^{(2)}(-s;-t) \succeq 0 = X^{(2)}(-t;-t)$. On the other hand, $Y^{\infty}(-s) \succeq X^{(1)}(-s;-s)$. By monotonicity and using the coupling events, at time $-t$, $Y^{\infty}(-t) \succeq X^{(1)}(-s;-t)$. This implies $X^{(1)}(-t;-t) = Y^{\infty}(-t) \wedge \mathbf{C} \succeq X^{(1)}(-s;-t)$. Therefore, at time $-t$, the trajectories starting further in the past at time $-s$ are sandwiched by those starting at time $-t$. This imposes coalescence for all the trajectories, starting at time $-s$ at the same point, at time 0, by monotonicity. The common value at coalescence is denoted Z in the rest of the proof.

We will show that $\mathbb{P}(Z = \mathbf{x}) = \pi_{\mathbf{x}}$, the stationary probability of state $\mathbf{x}$ for the chain $X(t)$. The proof is similar to the classical proof of the perfect sampling algorithm (see for example [Propp

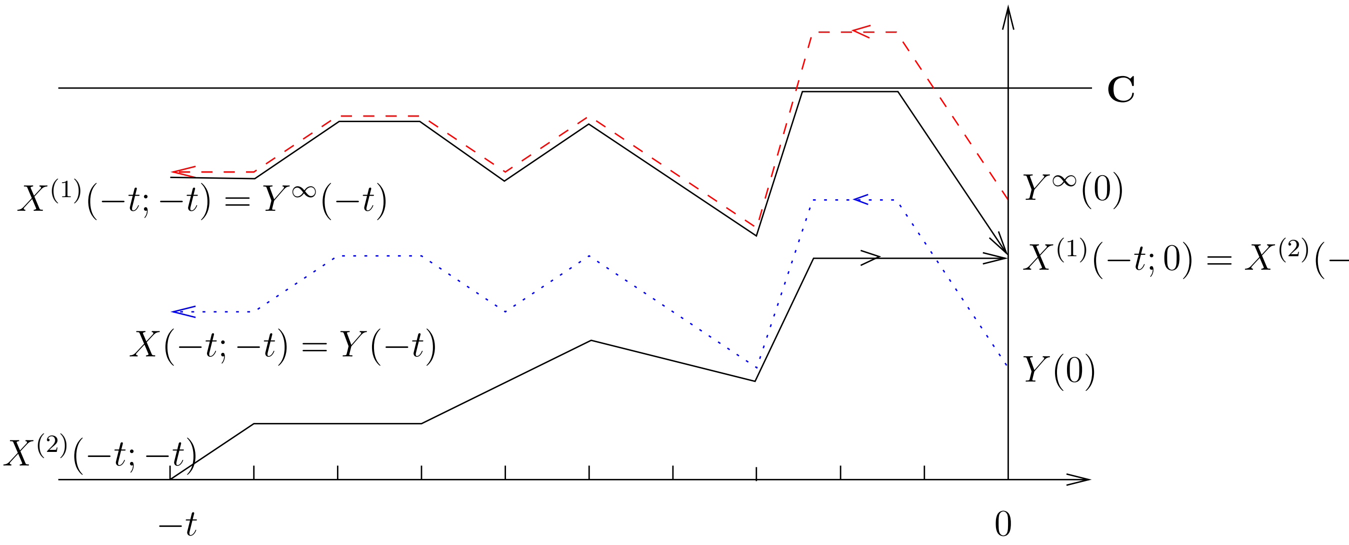


Figure 3: Illustration of the proof. The variable $Y(-t)$ has the stationary distribution of $X(t)$ and is below $Y^{\infty}(-t)$. For clarity, the forward trajectory $X(-t;\cdot)$ starting from $Y(-t)$ is not displayed. This trajectory would not exactly follow back the blue dotted trajectory of $Y(-t)$, unlike the upper coupled trajectories Y^{∞} and $X^{(1)}(-t;\cdot)$. The forward trajectory $X(-t;\cdot)$ will remain between the two extremas $X^{(1)}(-t;\cdot)$ and $X^{(2)}(-t;\cdot)$ and therefore end up in stationary state $X^{(1)}(-t;0)$ instead of state $Y(0)$.

and Wilson, 1996]). Let $0 < \epsilon < 1$. Since coalescence occurs in finite time almost surely, there exists T_ϵ such that $\mathbb{P}(\text{no coalescence at time } -T_\epsilon) < \epsilon$. Since coalescence implies that $X(-T_\epsilon;0) = Z$, we get

$$\mathbb{P}(Z \neq X(-T_\epsilon;0)) \leqslant \mathbb{P}(\text{no coalescence at } -T_\epsilon) < \epsilon.$$

On the other hand, the construction of $X(-T_\epsilon;0)$ makes sure that it has the stationary distribution π. Therefore,

$$\begin{aligned} \mathbb{P}(Z = \mathbf{x}) - \pi_{\mathbf{x}} &= \mathbb{P}(Z = \mathbf{x}) - \mathbb{P}(X(-T_\epsilon;0) = \mathbf{x}) \\ &\leqslant \mathbb{P}(Z = \mathbf{x}, X(-T_\epsilon;0) \neq \mathbf{x}) \\ &\leqslant \mathbb{P}(Z \neq X(-T_\epsilon;0)) < \epsilon. \end{aligned}$$

Similarly,

$$\begin{aligned} \pi_{\mathbf{x}} - \mathbb{P}(Z = \mathbf{x}) &= \mathbb{P}(X(T_\epsilon;0) = \mathbf{x}) - \mathbb{P}(Z = \mathbf{x}) \\ &\leqslant \mathbb{P}(Z \neq \mathbf{x}, X(-T_\epsilon;0) = \mathbf{x}) \\ &\leqslant \mathbb{P}(X(-T_\epsilon;0) \neq Z) < \epsilon. \end{aligned}$$

This is true for all ϵ so that $\mathbb{P}(Z = \mathbf{x}) = \pi_{\mathbf{x}}$.

□ □

3.2 When the infinite JQN is unstable

The Jackson network $\mathcal{J}$ admits a steady-state distribution (otherwise, perfect sampling does not make any sense) However, this may not be the case for $\mathcal{J}^{\infty}$: replacing bounded queues by infinite queues may result in an unstable infinite-queue JQN, for which no stationary distribution exists. In the previous sections, we have assumed that $\mathcal{J}^\infty$ is also stable. Here, we show that one can still construct an adequate bounding process for Algorithm 1 when $\mathcal{J}^\infty$ is not stable. Intuitively, the key idea is to determine the set $\mathcal{G}$ of queues that are unstable in $\mathcal{J}^\infty$ and to replace them with exogenous sources of packets (*a.k.a.* queue 0), with rate μ_i, $i \in \mathcal{G}$. Then we still have a stable infinite-queue JQN $\mathcal{J}^{\text{stab}}$ which can easily be sampled and reversed in time, only with a smaller number of queues. The state of the finite queues can then be bounded by their capacity C_i in the forward step of the algorithm. This trick will only additionally require to determine the set $\mathcal{G}$.

Let us consider the JQN $\mathcal{J}^\infty$. The traffic equations (for the stable as well as the unstable case) are:

$$\begin{cases} \forall i \neq 0, & \lambda_i = \sum_{j=1}^{M} p_{ji}(\lambda_j \wedge \mu_j) \\ \lambda_0 & = \mu_0 \end{cases} \tag{7}$$

where λ_i denotes the total incoming traffic rate in queue i. The set $\mathcal{G}$ of unstable queues is the set of queues such that $\lambda_i = \mu_i$. Solving the traffic equations was first done in [Goodman and Massey, 1984], where a proof of the uniqueness of the solution is provided along with an algorithm in $\mathcal{O}(M^4)$ to compute it. In appendix A we provide an alternative (shorter) proof of the existence and uniqueness (Theorem 5), and we show how it can be computed with complexity $\mathcal{O}(M^3)$ using Algorithm 4 and Proposition 3.

Now, the unique asymptotic distribution in the Jackson network $\mathcal{J}^\infty$ is still product form, and the marginal probabilities are given by:

- $\mathbb{P}(X_i = x) = (1 - \rho_i)\rho_i^x$ if i is stable,
- $\mathbb{P}(X_i = \infty) = 1$, if i is unstable.

It can be sampled in the same way as in the stable case.

The construction of the reversed Markov chain is also similar to the stable case, by extending the function β^∞ to take into account infinite values: an arrival or a departure in an unstable queue (with an infinite number of packets) will not change its size: $\infty + 1 = \infty - 1 = \infty$.

Therefore, Algorithm PSA-BP can also deal with unstable queues in $\mathcal{J}^\infty$ without any modification, once the unstable queues have been identified and their distribution in $\mathcal{J}^\infty$ correctly sampled, and once arithmetic operations have been overloaded to work with ∞ values.

Remark 2. *In the case where all queues have finite capacities and are unstable in $\mathcal{J}^\infty$, Algorithm PSA-BP degenerates to a classical PSA, since the starting points of the forward simulation will be $(0, 0, \ldots, 0)$ and $(C_1, C_2, \ldots, C_M)$. Indeed, in that case, the bounding process Y^∞ is trivial, and has all its coordinates equal to ∞.*

One way to avoid this situation, is to change the variables to be sampled. Instead of sampling X_i the number of packets in the queue i, it is better to sample the variables $\overline{X}_i \overset{\text{def}}{=} C_i - X_i$, the number of empty slots in queue i. Doing so, the network $\mathcal{J}$ where all queues are unstable is transformed into a Jackson network $\overline{\mathcal{J}}$ where all queues are stable, for which the bounding process will not be trivial and will improve the simulation time.

In general (some queues are stable and some others are unstable), one can simulate either $\mathcal{J}$ or $\overline{\mathcal{J}}$. The choice can be based on the asymptotic complexity of the algorithm PSA-BP, given below, that depends almost exclusively on the asymptotic distribution of $\mathcal{J}^\infty$ (resp. $\overline{\mathcal{J}}^\infty$).

4 Complexity analysis

The simulation time of PSA-BP can be decomposed into three steps:

1. The generation of a sample $Y^\infty(0)$ with the stationary distribution of the bounding process. Since this distribution is product form and since for each queue the geometric law can be easily sampled in constant time, the time complexity of this first step is $\mathcal{O}(M)$.

2. To obtain random events from the innovation sequence $\{U_{-n}\}_{n\in\mathbb{N}}$, we use the Walker's alias method [Walker, 1977] (for sampling discrete random variables in $\mathcal{O}(1)$ time). The construction of alias table is linear in number of events, so this pre-treatment can be done in $\mathcal{O}(M^2)$ time. Given the events r_{ij}, the values of $\phi(\mathbf{x}, r)$ and $\beta^\infty(\mathbf{x}, r)$ can be computed in $\mathcal{O}(1)$ time.

3. The construction of the backward trajectory of Y^∞ and the forward trajectories of $X^{(1)}$ and $X^{(2)}$ have the same expected duration, smaller than $10\mathbb{E}\tau(X^{(2)}, X^{(1)})$, where $\tau(X^{(2)}, X^{(1)})$ is the coalescence time of two trajectories of the Markov chain $X(t)$, starting in $X^{(1)}$ and $X^{(2)}$ respectively, under the same sequence of events. The multiplication factor 10 comes from a simple calculation: 3 trajectories, each with length at most double the coalescence time (worst case), and two of them (for $X^{(1)}$ and $X^{(2)}$) requiring a computation time equal to 2τ to compute a τ-long trajectory ($1 + 2 + 4 + \ldots + \tau/2 + \tau \approx 2\tau$ steps).

Therefore, the complexity essentially depends linearly on the coalescence time $\tau(X^{(2)}, X^{(1)})$.

4.1 The acyclic case

In the following, we will show that in the acyclic case, the coalescence time can be bounded by a quadratic function of the expected size of the queues in the stationary regime. Simulations (see Section 6) suggest that this bound is tight and holds for all stable networks, including cyclic ones.

Theorem 2. *If $\mathcal{J}$ is acyclic and stable, then the expected coalescence time of algorithm PSA-BP, $\tau_{\text{PSA-BP}}$ satisfies*

$$\mathbb{E}[\tau_{\text{PSA-BP}}] \leqslant \sum_{i=1}^{M} \frac{\Gamma}{\lambda_i} (\mathbb{E}[Y_i^\infty])^2, \tag{8}$$

where μ_i is the service rate in queue i, λ_i is the input rate in queue i, $\mathbb{E}[Y_i^\infty] = \frac{\lambda_i}{\mu_i - \lambda_i}$ is the expected stationary size of queue i of the infinite system $\mathcal{J}^\infty$, and Γ is the uniformization constant, $\Gamma = \sum_i (\mu_i + \lambda_i)$.

Remark 3. *The expected coalescence time grows as the square root of the sizes of the queues when the load increases, since the multiplicative coefficients $\frac{\Gamma}{\lambda_i}$ remain bounded when the load goes to 1.*

The proof is carried in several steps.

Lemma 3. *Let $T(s)$ be the expected time needed to empty a stable queue in discrete time with arrival (resp. service) probability $\frac{\lambda}{\lambda+\mu}$ (resp. $\frac{\mu}{\lambda+\mu}$), starting with s customers. Then $T(s) = s\frac{\lambda+\mu}{\mu-\lambda}$.*

Proof. (of the lemma). This can be considered as a classical result, the proof is given nevertheless. $T(s)$ satisfies the following one-step recurrence: $T(s) = 1 + \frac{\lambda}{\lambda+\mu} T(s+1) + \frac{\mu}{\lambda+\mu} T(s-1)$, whose solution is of the form $T(s) = C_1 \left(\frac{\mu}{\lambda}\right)^s + C_2 s$. The constants C_2 and C_1 are functions of $T(0)$ and $T(1)$. By definition, $T(0) = 0$ and $T(1)$ is such that $C_1 = 0$ (otherwise the queue cannot be stable). This implies $T(1) = \frac{\lambda+\mu}{\mu-\lambda}$. and $T(s) = s\frac{\lambda+\mu}{\mu-\lambda}$.

Another proof: The quantity $T(x)$ satisfies the recurrence: $T(x) = BP + T(x-1)$, where BP is the expected busy period of the slotted queue, known to be equal to $\frac{\lambda+\mu}{\mu-\lambda}$. The fact that $T(0) = 0$ implies that $T(x) = x\frac{\lambda+\mu}{\mu-\lambda}$.

□ □

Let us consider the reversed network $\mathcal{K}^{\infty}$. Since $\mathcal{J}$ is stable and acyclic, $\mathcal{K}^{\infty}$ is also stable and acyclic. The queues are ordered according to the topological order induced by the network $\mathcal{K}^{\infty}$. In each queue i of $\mathcal{K}^{\infty}$, the service rate is μ_i and the input rate is λ_i, defined by the linear system $\lambda_i = \sum_j \lambda_j q_{j,i}$ under its stationary regime.

The stability of $\mathcal{K}^{\infty}$ can be translated as for all i, $\lambda_i < \mu_i$. Now, for any initial state x in $\mathcal{K}^{\infty}$, let us define the time $h(x)$ as follows: $h(x)$ is the first instant when the following sequence of events have occurred, in that order: Queue 1 hits 0 then Queue 2 hits 0 then ... then Queue M hits 0. The stability of $\mathcal{K}^{\infty}$ implies that $h(x)$ is finite, with a finite expectation, for all x.

Lemma 4. *The expectation $\mathbb{E}[h(Y^{\infty}(0))]$, where $Y^{\infty}(0)$ has the stationary distribution of $\mathcal{K}^{\infty}$, is linear:*

$$\mathbb{E}[h(Y^{\infty}(0))] = \sum_i \frac{\Gamma}{\mu_i - \lambda_i}\mathbb{E}[Y_i^{\infty}(0)].$$

Proof. (of the lemma).

The proof holds by induction on the number of queues. The case $M = 1$ follows directly from Lemma 3:

$$\mathbb{E}[h(Y^{\infty}(0))] = \mathbb{E}[T(Y^{\infty}(0))] = \frac{\lambda+\mu}{\mu-\lambda}\mathbb{E}[Y^{\infty}(0)]$$

and $\Gamma = \lambda + \mu$.

Let us consider an acyclic network with M queues and let us cut the last queue (Queue M). The time $h(Y^{\infty}(0))$ is equal to the time when the first $M-1$ queues reach 0 plus the time for the last queue to become empty from that point on. By Burke's Theorem, when the initial state S is stationary, then the state of queue M at any time is independent of the state of the other queues. Therefore, at hitting times of the other queues, the state of queue M is distributed according to the stationary distribution, and the input process in queue M is Poisson with rate λ_M.

Let us call Y_M the state of Queue M just after the first $M-1$st queues have emptied (in that order). Its distribution is $\mathbb{P}(Y_M = i | Y_{M-1} = 0) = \mathbb{P}(Y_M = i)$, by independence of the queues under the stationary regime, so that its distribution is stationary. From that point on, Queue M evolves as an independent queue with arrival rate λ_M and service rate μ_M. Using Lemma 3 the expected time to hit 0 for this last queue, under uniformization Γ, from that time on is:

$$\frac{\Gamma}{\mu_M - \lambda_M}\mathbb{E}[Y_M^{\infty}(0)].$$

As for the $M-1$ first queues, the induction assumption says that the expected time for all queues to hit 0 is

$$\sum_{i=1}^{M-1} \frac{\Gamma}{\mu_i - \lambda_i}\mathbb{E}[Y_i^{\infty}(0)].$$

Adding both quantities gives $\mathbb{E}[h(Y^{\infty}(0))] = \sum_{i=1}^{M} \frac{\Gamma}{\mu_i-\lambda_i}\mathbb{E}[Y_i^{\infty}(0)]$.

□ □

We are now ready for the proof of Theorem 2.

Proof. (of Theorem 2).

Let us consider the state reached by the reversed network, starting in state $Y^\infty(0)$, at time $-h(Y^\infty(0))$, namely $Y^\infty(-h(Y^\infty(0)))$ (for simplicity, we will denote this hitting time by $h^* \overset{\text{def}}{=} h(Y^\infty(0))$).

Now, let us consider the initial network $\mathcal{J}$ starting at time $-h^*$ in state $Y^\infty(-h^*) \wedge \mathbf{C}$. Because of the coupling between the reversed and direct sequences of events, the trajectory $X^{(1)}(-h^*; -t)$ remains below (componentwise) the reversed trajectory, $Y^\infty(-t)$ for any t. This means that the following sequence of events must occur in the trajectory $X^{(1)}(-h^*; -t)$:
queue M hits 0, then queue $M-1$ hits 0, ... , queue 1 hits 0.

Let us now consider the direct trajectory starting in 0 at time $-h^*$, namely $X^{(2)}(-h^*; -t)$. By monotonicity, for all t, $X^{(2)}(-h^*; -t) \preceq X^{(1)}(-h^*; -t)$. Therefore, when queue M hits 0 in $X^{(1)}(-h^*; -t)$, then queue M also hits 0 in $X^{(2)}(-h^*; -t)$ at the same instant. This means that the state of queue M is the same on both trajectories. By acyclicity, from that point on, the M-th coordinate of $X^{(2)}(-h^*; -t)$ and $X^{(1)}(-h^*; -t)$ will remain equal. When queue $M-1$ hits 0, the same phenomenon will happen: The $M-1$-st coordinate of $X^{(2)}(-h^*; -t)$ and $X^{(1)}(-h^*; -t)$ will remain equal from that point on, and so forth. This implies that the two trajectories must coalesce before time 0.

This means that the forward coalescence time $\tau(X^{(2)}, X^{(1)})$ is smaller than $h^* = h(Y^\infty(0))$. Using Lemma 4, if $Y^\infty(0)$ is stationary, then the expected coalescence time $\mathbb{E}\tau(X^{(2)}, X^{(1)})$ satisfies the required inequality:

$$\mathbb{E}[\tau_{\text{PSA-BP}}] \leqslant \mathbb{E}[h(Y^\infty(0))] = \sum_{i=1}^{M} \frac{\Gamma}{\mu_i - \lambda_i} \mathbb{E}[Y_i^\infty(0)] = \sum_{i=1}^{M} \frac{\Gamma}{\lambda_i} \big(\mathbb{E}[Y_i^\infty(0)]\big)^2.$$

This concludes the proof.

□

□

4.2 The hyperstable case

In this section, we consider the case where the network has an arbitrary topology (that may contain cycles) but where all queues are hyperstable (to be defined later).

In this case we also obtain a bound on the expected coupling time. This new bound is cubic in the average queue size (instead of quadratic in the acyclic case). Notice that this bound remains logarithmic with respect to the size of the state space. This shows that the algorithm PSA-BP remains very efficient even for cyclic networks.

Let $\tau_{\text{PSA-BP}}$ be the coalescence time of algorithm PSA-BP. It is a finite random variable with finite expectation when the network is stable.

On the other hand, let us define $\tau_f(\mathbf{x}, \mathbf{y})$ the meeting time of two trajectories of the Jackson network $\mathcal{J}$, starting in states $\mathbf{x}$ and $\mathbf{y}$ respectively. This is also a finite random variable with finite expectation when the network is stable.

The next lemma is a quite general result, not based on hyper stability.

Lemma 5. *Under the foregoing assumptions, if $\mathcal{J}$ is stable, then $\tau_f(\mathbf{0}, Y^\infty(0) \wedge \mathbf{C})$ and $\tau_{\text{PSA-BP}}$ have the same distribution.*

Proof. Let T be a fixed duration. Using the notations introduced in the proof of Theorem 1,

$$\begin{aligned}
\mathbb{P}(\tau_{\text{PSA-BP}} < T) &= \mathbb{P}(X^{(1)}(-T;0) = X^{(2)}(-T;0)) &(9)\\
&= \mathbb{P}(\phi(\mathbf{0}, v_{-T+1}, \dots, v_0) = \phi(Y^\infty(-T) \wedge \mathbf{C}, v_{-T+1}, \dots, v_0)) &(10)\\
&= \mathbb{P}(\phi(\mathbf{0}, v_{-T+1}, \dots, v_0) = \phi(Y^\infty(0) \wedge \mathbf{C}, v_{-T+1}, \dots, v_0)) &(11)\\
&= \mathbb{P}(\phi(\mathbf{0}, u_1, \dots, u_T) = \phi(Y^\infty(0) \wedge \mathbf{C}, u_1, \dots, u_T)) &(12)\\
&= \mathbb{P}(\tau_f(\mathbf{0}, Y^\infty(0) \wedge \mathbf{C}) < T), &(13)
\end{aligned}$$

where $u_1, \dots, u_T$ is a sequence of *iid* events such that for all t, $\mathbb{P}(u_t = r_{ij}) = a_{ij}$. Equation (11) comes from the fact that for any T, $Y^\infty(T)$ is stationary, Equation (12) is true because the sequences $u_1, \dots, u_T$ and $v_{-T+1}, \dots v_0$ have the same distribution by Property 2 and the last equation (13) follows from the definition of τ_f.

□ □

Now, let us now consider the hyper-stable case.

Definition 1. *A queue $j \geq 1$ in a Jackson network $\mathcal{J}$ is hyper-stable if $\sum_{i=0}^M \gamma_{ij} < \mu_j$. The network is hyper-stable if all queues $j \geq 1$ are hyper-stable. Note that hyper-stability implies stability since $\lambda_j \leqslant \sum_{i=0}^M \gamma_{ij}$, $\forall j \geq 1$.*

Theorem 3. *If $\mathcal{J}$ is hyper-stable then the expected coalescence time of algorithm PSA-BP satisfies $\mathbb{E}\tau_{\text{PSA-BP}} \leqslant cM\Gamma \max_i \left(\frac{\mathbb{E}Y_i^\infty}{\lambda_i}\right) (\sum_{j=1}^M \mathbb{E}Y_j^\infty)^2$, where c is a constant (that only depends on the routing probabilities (p_{ij})) and $\mathbb{E}Y_j^\infty$ is the expected stationary size of queue j of the infinite system $\mathcal{J}^\infty$ ($\mathbb{E}Y_i^\infty = \frac{\lambda_i}{\mu_i - \lambda_i}$).*

Proof. Using Lemma 5, we will consider $\tau_f(\mathbf{0}, Y^\infty(0) \wedge \mathbf{C})$ instead of $\tau_{\text{PSA-BP}}$ since they have the same distribution. Now, in [Anselmi and Gaujal, 2013], it was proved (proof of Theorem 2 in [Anselmi and Gaujal, 2013]) that for hyperstable Jackson networks with M queues, there exists a constant c such that for all i and for any state $\mathbf{y}$,

$$\mathbb{E}\tau_f(\mathbf{0}, \mathbf{y} + \mathbf{e_i}) - \mathbb{E}\tau_f(\mathbf{0}, \mathbf{y}) \leqslant cM\Gamma \max_i \left(\frac{\mathbb{E}Y_i^\infty}{\lambda_i}\right) \sum_{j=1}^M \frac{\lambda_j}{\mu_j - \lambda_j}.$$

By iterating this inequality and conditioning on the value of $Y^\infty(0)$, one gets

$$\mathbb{E}[\tau_f(\mathbf{0}, Y^\infty(0)) | Y^\infty(0)] \leqslant cM\Gamma \max_i \left(\frac{\mathbb{E}Y_i^\infty}{\lambda_i}\right) \left(\sum_{j=1}^M \frac{\lambda_j}{\mu_j - \lambda_j}\right) \sum_i Y_i^\infty(0).$$

Finally,

$$\begin{aligned}
\mathbb{E}[\tau_f(\mathbf{0}, Y^\infty(0) \wedge \mathbf{C})] &\leqslant \mathbb{E}[\tau_f(\mathbf{0}, Y^\infty(0))]\\
&\quad = \mathbb{E}\Big[\mathbb{E}[\tau_f(\mathbf{0}, Y^\infty(0)] | Y^\infty(0)]\Big]\\
&\leqslant Mc\Gamma \max_i \left(\frac{\mathbb{E}Y_i^\infty}{\lambda_i}\right) \left(\sum_{j=1}^M \frac{\lambda_j}{\mu_j - \lambda_j}\right) \sum_i \mathbb{E}Y_i^\infty(0)\\
&\quad = Mc\Gamma \max_i \left(\frac{\mathbb{E}Y_i^\infty}{\lambda_i}\right) \left(\sum_{j=1}^M \mathbb{E}Y_i^\infty\right)^2.
\end{aligned}$$

□ □

5 Comparison with the classical perfect sampler

As discussed earlier, our new algorithm allows for perfect sampling of Jackson networks with finite and infinite buffers. This was not possible with the classical perfect sampling algorithms that requires a finite state space. As such, our algorithm broadens the scope of perfect sampling techniques. In addition, this new approach also provides time improvements in the case where all buffers are finite, because the time complexity of the new approach is essentially independent of the capacities. In that case, it reduces the sampling time by a factor corresponding to the ratio between the maximum capacity of the buffers over the expected size of the queues under the stationary law of the bounding process, at least in the acyclic case (the comparison in the cyclic case is not as striking because of the squared term but behaves essentially in the same way).

Let us first recall the classical perfect sampler for monotone finite Markov chains derived from Jackson queuing networks.

Algorithm 2: Coupling from the past (PSA) [Propp and Wilson, 1996]

```
Data: An infinite i.i.d. sequence $\{U_{-n}\}_{n\in\mathbb{N}}$ with $U_{-k}$ uniformly distributed in $[0,1]$
      (with the associated sequence of events $\{u_{-n}\}_{n\in\mathbb{N}}$).
Result: A state sampled from the stationary distribution of the JQN
begin
    $t := 1$;
    repeat
        $Z^{(1)} := \mathbf{C}$; $Z^{(2)} := 0$;
        for $i = t-1$ downto $0$ do
            $Z^{(1)} := \phi(Z^{(1)}, u_{-i})$;
            $Z^{(2)} := \phi(Z^{(2)}, u_{-i})$;
        $t := 2t$;
    until $Z^{(1)} = Z^{(2)}$ ;
    return $Z^{(1)}$;
end
```

It is well known that Algorithm 2 outputs samples of the stationary distribution of the network as long as all buffers are finite. Otherwise, if $X^{(1)}$ has at least one infinite coordinate, the algorithm does not converge.

Also note that, in the general case (acyclic or cyclic), the coalescence time of PSA-BP is always stochastically smaller than the coalescence time of PSA since the extreme starting point $X^{(1)}(-t) = Y^{\infty}(-t) \wedge \mathbf{C}$ of the former is always smaller than the starting time of the latter, $Z^{(1)}(-t) = \mathbf{C}$.

Finally, the time complexity in the case where $\mathcal{J}$ is acyclic with finite queues has been studied in [Dopper et al., 2006]. The expected coalescence time to get one sample τ_{PSA} is shown to satisfy

$$\mathbb{E}[\tau_{\text{PSA}}] \leqslant \sum_{i=1}^{M} \frac{\Gamma}{\mu_i + \lambda_i}(C_i^2 + C_i).$$

It is also easy to find a lower bound on the expected coalescence time:

$$\mathbb{E}[\tau_{\text{PSA}}] \geqslant \sum_{i=1}^{M} \frac{\Gamma}{\mu_i + \lambda_i} C_i.$$

Indeed, one needs at least C_i departures in queues i for coalescence and the average time between two events in queue i is $\frac{\Gamma}{\mu_i+\lambda_i}$.

As shown in Theorem 2, the expected coalescence time of our new sampler is upper bounded:

$$\mathbb{E}[\tau_{\text{PSA-BP}}] \leqslant \sum_{i=1}^{M} \frac{\Gamma}{\lambda_i} \mathbb{E}[Y_i^\infty]^2$$

(where $\mathbb{E}Y_i^\infty$ is the expected stationary size of queue i in the infinite system $\mathcal{J}^\infty$). This may result in a considerable improvement over the classical algorithm.

Indeed, the upper and lower bounds above induce the following comparison between the coalescence time of the two algorithms:

$$\mathbb{E}[\tau_{\text{PSA}} - \tau_{\text{PSA-BP}}] \geqslant \max\left(0, \sum_{i=1}^{M} \Gamma\left(\frac{\mathbf{C}_i}{\lambda_i+\mu_i} - \frac{\mathbb{E}[Y_i^\infty]^2}{\lambda_i}\right)\right).$$

If the arrival rate is 0.8 in every queue with service rate 1, (this is a typical situation), then $\mathbb{E}[Y_i^\infty] \leqslant 4$, and each term in the difference is larger than $\mathbf{C}_i/2 - 20$ which can be very large when the buffer size of the order of 100, 1000 or more.

In the next section, we show on a numerical example that the PSA-BP algorithm 1 drastically outperforms the classical PSA Algorithm 2.

6 Numerical experiments

We have implemented algorithms 1 and 2 (PSA-BP and PSA) in C, in order to:

- demonstrate the feasibility of our approach,
- quantify the actual CPU time speedup of our approach over classic PSA,
- and estimate the tightness of the complexity bounds derived in section 4.

Experimental setup

We tested both simulation algorithms over a circular queuing network displayed in Figure 4 : the M queues form a cycle, with the same service time in each queue, $\mu_i = \mu$, the same exogenous arrival rate in each queue (λ) and the same probability of leaving the system at each queue: $p_{i0} = p$. In that case, the infinite-queue system is stable if and only if $\lambda/p < \mu$. To be consistent with the notation in Table 1, we have $\lambda_0 = M \times \lambda$.

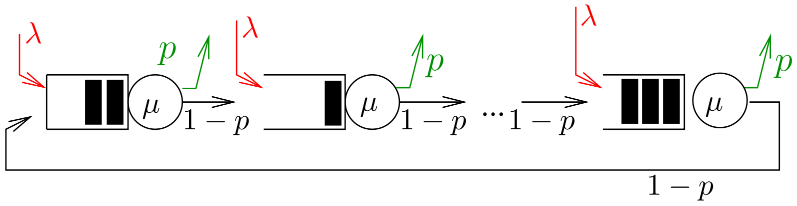


Figure 4: Circular network $\mathcal{J}$ used in the numerical experiments.

We have run both simulators for various queue capacities $C_i = C$ for $1 \leqslant i \leqslant M$ ranging from 1 to 600, and various traffic loads $\rho_i = \rho$ ranging from 0.6 to .99.

Our experiments were carried out on a standard laptop computer[2]. All results are given with 95 % confidence intervals computed with 300 experiments per setting.

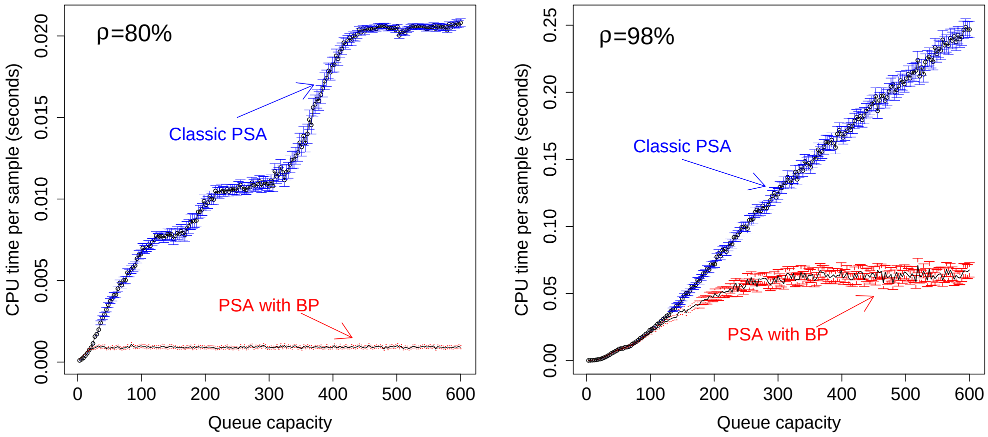
6.1 Sampling time improvement over classic PSA

The complexity analysis in Section 4 shows that the coalescence time of PSA-BP is always smaller, however we first wanted to verify that

- the practical sampling time (CPU time) is well predicted by the coalescence time,
- the CPU time is significantly better (the overhead due to the construction of the bounding process remains negligible).

To this end we measured the running time of both algorithms (PSA-BP and classic PSA) for the same circular network, over a wide range of queue capacities.

Figure 5 presents the CPU time [3] needed per run for both PSA and PSA-BP algorithms, under average (Fig. 5(a)) and high loads (Fig. 5(b)). The capacity C of the queues ranges from 1 to 600. These capacity choices are to be compared to the expected stationary queue size of the bounding infinite-queue network, which is $\frac{\rho}{1-\rho}$, i.e. 4 jobs (resp. 49 jobs) for a 80% load (resp. 98%).



(a) Low traffic intensity ($\lambda = 0.4$) $\mathbb{E}[Y^\infty] = 4$ jobs (b) High traffic intensity ($\lambda = 0.49$) $\mathbb{E}[Y^\infty] = 49$ jobs

Figure 5: Comparative sampling time (CPU time consumption per sample) of Bounding-Process PSA and Classic PSA. Circular networks with M=3 queues, $\mu = 1, p_{i0} = 0.5$. The capacity C of each queue varies from 1 to 600. 95% confidence intervals with 300 samples per setting.

We observe that the simulation time of the classical perfect sampling algorithm grows somewhat linearly with the capacity of the buffers, as does the coalescence time (not displayed here to avoid redundancy). The succession of plateaus and steep slopes (staircase shape) comes from

[2] Intel Core 2 Duo CPU U9400 1.40GHz, Cache 3072KB with 4.8GB RAM. Only one core is used for each sample.

[3] CPU time used by the process computing one sample, measured with the CLOCK_PROCESS_CPUTIME_ID of the time.h library.

the doubling period trick of the perfect simulation algorithm: the coalescence time of Algorithm 2 is the smallest power of 2 larger than the actual coalescence time of the trajectory. This linear increase behavior, while the state space grows with C^M, is consistent with the findings in [Anselmi and Gaujal, 2011].

As for the proposed algorithm 1, Figure 5 perfectly illustrates the fact that the CPU sampling time remains bounded when the state space increases, as shown for the expected coalescence time by Theorem 2 in the acyclic case.

The performance gap with the classical algorithm becomes significant as soon as the capacity becomes larger than a few times the bounding expected stationary queue size ($C \geqslant 10$ for Figure 5(a), $C \geqslant 200$ for Figure 5(b)). Indeed, when C is small, the bounding infinite-queue process is too large and algorithm PSA-BP roughly amounts to classic PSA, except for the computation of the bounding trajectory. Even in this unfavorable case, we do not observe any significant overhead effect.

6.2 Complexity bound

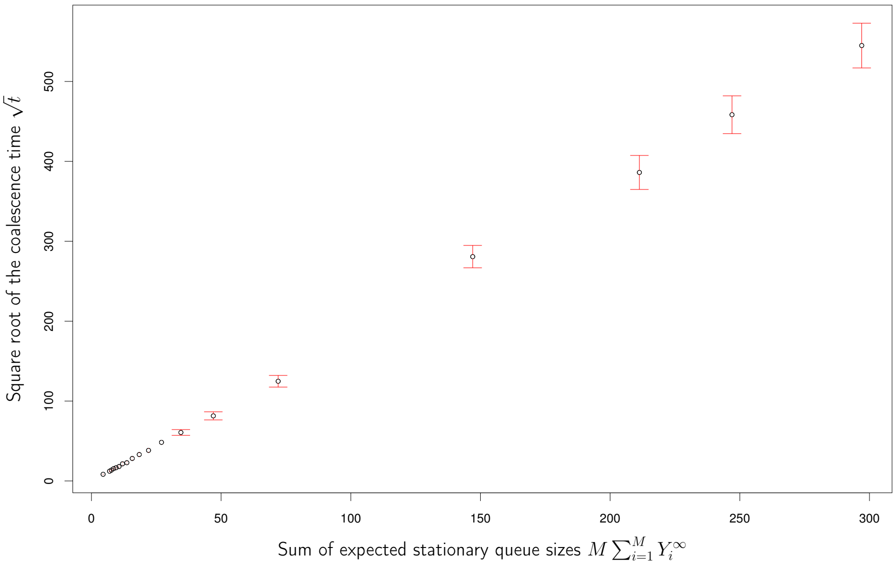


Figure 6: Asymptotic behavior of (the square root of) the coalescence time (number of steps) of our PSA-BP algorithm as the expected queue size grows (increasing with load ρ). Notice that due to the doubling period artefact, the observed number of steps is at most twice the actual coalescence time of the simulated trajectories. Circular networks with M=3 queues, capacity $C = 500, \mu = 1, p_{i0} = 0.5$.

In Theorems 2 and 3, upper bounds on the coalescence time are given in the acyclic and hyper-stable cases, that are respectively quadratic and cubic in the expected size of the queues for the bounding network $\mathcal{J}^\infty$.

On the other hand, it is rather straightforward to show that for any stable network $\mathcal{J}$, the coalescence time is lower bounded by a sub-linear combination of the expected sizes of the queues in the network:

$$\mathbb{E}[\tau_{\text{PSA-BP}}] \geqslant \sum_{i=1}^{M} \frac{\Gamma}{\mu_i + \lambda_i} \mathbb{E}[Y_i^\infty \wedge C_i].$$

Indeed, the coalescence time in the network has the same distribution as the forward coalescence time $\tau_f(\mathbf{0}, \mathbf{Y}^\infty \wedge \mathbf{C})$ (see Lemma 5). One needs at least $Y_i^\infty \wedge C_i$ departures in queue i for coalescence and the average time between two events in queue i is $\frac{\Gamma}{\mu_i+\lambda_i}$. Therefore in expectation,

$$\mathbb{E}[\tau_f(\mathbf{0}, \mathbf{Y}^\infty \wedge \mathbf{C})] \geqslant \sum_{i=1}^{M} \frac{\Gamma}{\mu_i + \lambda_i} \mathbb{E}[Y_i^\infty \wedge C_i].$$

To test if the actual coalescence time behaves like the lower bound (linear) or the upper bound (quadratic or cubic), or does not comply with any of them in the cyclic non-hyperstable case, we have run several experiments by letting the load grow from 50 % to 98 % and we measured the coalescence time versus the average size of the queues in the circular network of Figure 4. The results are given in Figure 6. The behavior of the square root of the coalescence time follows almost perfectly the average size of the queues, up to a multiplicative constant. This shows a quadratic behavior of the coalescence time, even though the network is not acyclic (nor hyperstable). We conjecture that the bound given in Theorem 2 is actually true for any stable network and that it is tight.

7 Extension to non-monotone networks

The perfect sampling technique presented here can be extended to non-monotone chains that admit a monotone bounding chain, by combining Algorithm 1 with the envelope technique presented in [Busic et al., 2012b].

We illustrate this extension on the case of a network with negative customers [Gelenbe, 1991]. Let $\mathcal{N}$ be a network with finite and infinite buffer capacities and positive and negative customers: the arrival of a negative customer in a buffer decreases the buffer size by one (unless the buffer is empty), instead of increasing it by one for regular (positive) customers.

Denote by p_{ij}^+ (resp. p_{ij}^-) the probability that after a service in queue i the customer goes to queue j as a positive (resp. negative) customer. The discrete-event representation for this system can be obtained from the one for JQN, by dividing each event r_{ij} $(i, j \in \{0, 1, \dots, M\})$ into two new events, denoted by r_{ij}^+ and r_{ij}^- of rates $\gamma_{ij}^+ := \mu p_{ij}^+$ and $\gamma_{ij}^- := \mu p_{ij}^-$. The total event rate is $\Gamma \overset{\text{def}}{=} \sum_{i,j=0}^{M} (\gamma_{ij}^+ + \gamma_{ij}^-)$.

We assume here that the matrix $P^+ = \{p_{ij}^+\}_{i,j \in \{0,1,\dots,M\}}$ is irreducible. Furthermore, we assume positive stability, *i.e.*

$$\lambda_i < \mu_i, \ 1 \leqslant i \leqslant M, \tag{14}$$

where λ_i satisfy the equation $\lambda_j = \sum_{i=0}^{M} \lambda_i p_{ij}^+$.

As before, we consider a discrete-time Markov chain $\{Z(t)\}_{t\in\mathbb{N}}$, after uniformization with constant Γ. Using the assumption that the routing matrix P^+ is irreducible, this discrete chain is irreducible and aperiodic. and its evolution is given by: $Z(t+1) = \psi(Z(t), v_{t+1})$ where $(v_t)_{t\geqslant 0}$ are events (routing positive or negative customers, *i.e.* r_{ij}^+ or r_{ij}^-, $0 \leqslant i, j \leqslant M$. The forward transition function of the chain $\psi : \mathcal{S} \times \mathcal{R} \to \mathcal{S}$ is defined as follows:

- $\psi(\mathbf{x}, r_{ij}^+) = ((\mathbf{x} - \mathbf{e}_i) \vee 0 + \mathbf{e}_j \mathbb{1}_{\mathbf{x}-\mathbf{e}_i \geqslant 0}) \wedge \mathbf{C}$;
- $\psi(\mathbf{x}, r_{ij}^-) = (\mathbf{x} - \mathbf{e}_i - \mathbf{e}_j \mathbb{1}_{\mathbf{x}-\mathbf{e}_i \geqslant 0}) \vee 0$;
- $\psi(\mathbf{x}, r_{0j}^+) = (\mathbf{x} + \mathbf{e}_j) \wedge \mathbf{C}$;
- $\psi(\mathbf{x}, r_{0j}^-) = (\mathbf{x} - \mathbf{e}_j) \vee 0$;

- $\psi(\mathbf{x}, r^+_{i0}) = (\mathbf{x} - \mathbf{e}_i) \vee 0$.
- $\psi(\mathbf{x}, r^-_{i0}) = (\mathbf{x} - \mathbf{e}_i) \vee 0$.

Events r^-_{ij} are not monotone for $1 \leqslant i, j \leqslant M$. To see this, consider the states $x_1 = \mathbf{e}_j$ and $x_2 = \mathbf{e}_i + \mathbf{e}_j$. The inequality $x_1 \prec x_2$ holds. By definition, $\psi(x_1, r^-_{ij}) = \mathbf{e}_j$ and $\psi(x_2, r^-_{ij}) = \mathbf{0}$, so $\psi(x_2, r^-_{ij}) \prec \psi(x_1, r^-_{ij})$.

It is nevertheless possible to construct a new network $\mathcal{J}^\infty$ with infinite buffers in all queues and where negative customers are replaced by departures (events r^-_{ij} are replaced by events r^+_{i0}). Network $\mathcal{J}^\infty$ corresponds to a classical Jackson network with routing probabilities $p_{ij} := p^+_{ij}$, for $0 \leqslant i \leqslant M$ and $1 \leqslant j \leqslant M$, and $p_{i,0} := p^+_{i,0} + \sum_{j=0}^{M} p^-_{ij}$. The state of $\mathcal{J}^\infty$ is a Markov chain whose forward transition function is denoted ϕ^∞. Under assumption (14), network $\mathcal{J}^\infty$ is stable. The network $\mathcal{J}^\infty$ is also clearly monotone and with the coupling of negative customer routing with departures, it is an upper bound of $\mathcal{N}$, so Proposition 1 remains valid.

This allows us to construct a perfect sampler of network $\mathcal{N}$ using the envelope technique presented in [Busic et al., 2012b] for the forward part, and the product form network $\mathcal{J}^\infty$, with stationary measure π^∞, for the backward part. This perfect sampler is given in Algorithm 3, and the events for both the reversed network and the forward one are coupled using the following construction, similar to the coupling described in Proposition 2.

The reversed chain of network $\mathcal{J}^\infty$ has the same routing events r^+_{ij}, but different probabilities, whose corresponding intervals are denoted B_{ij}, as in Section 2.2.

Let $Y^\infty(0)$ by a stationary sample of the infinite reversed system (with no negative customers), generated according to the product form distribution π^∞. Given an infinite sequence of i.i.d. uniformly distributed random variables in $[0,1]$, $U_{-1}, U_{-2}, \cdots$, we construct an infinite sequence of events $w_{-1}, w_{-2}, \cdots$ of events for the reversed chain as follows. The intervals B_{ij} are ordered to cover $[0,1]$ using the alpha-numerical order:

$$B_{0,1}, B_{0,2}, \ldots B_{0,M}, B_{1,0}, B_{1,1}, \ldots, \ldots, B_{M,M}.$$

Now, if $U_{-t} \in B_{ij}$, then $w_{-t} = r^+_{ij}$.

Then, we construct a trajectory of the reversed chain starting from $Y^\infty(0)$ and using the sequence of events: for all $t > 0$, $Y^\infty(-t) = \phi^\infty(Y^\infty(-t+1), w_{-t})$.

Now, the coupled event sequence for the direct chain $v_{-1}, \ldots, v_{-t}, \ldots$ is built as described by the following coupling procedure (C_2):

1. If $Y^\infty(-t) \neq Y^\infty(-t+1)$ then
   - if $w_{-t} = r^+_{0,i}$ then $v_{-t} = r^+_{i,0}$ with probability $\frac{p^+_{i,0}}{p^+_{i,0} + \sum_{j=0}^{M} p^-_{ij}}$ and $v_{-t} = r^-_{ij}$ with probability $\frac{p^-_{ij}}{p^+_{i,0} + \sum_{j=0}^{M} p^-_{ij}}$.
   - if $w_{-t} = r^+_{ij}$ $(i \neq 0)$ then $v_{-t} = r^+_{j,i}$
2. If $Y^\infty(-t) = Y^\infty(-t+1)$ then event w_{-t} must be of the form $w_{-t} = r^+_{ij}$ with $i \neq 0$.
   - if $w_{-t} = r^+_{i,0}$ then $v_{-t} = r^+_{i,0}$ with probability $\frac{p^+_{i,0}}{p^+_{i,0} + \sum_{j=0}^{M} p^-_{ij}}$ and $v_{-t} = r^-_{ij}$ with probability $\frac{p^-_{ij}}{p^+_{i,0} + \sum_{j=0}^{M} p^-_{ij}}$.
   - if $w_{-t} = r^+_{ij}$ $(j \neq 0)$ then $v_{-t} = r^+_{ij}$

Using this coupling, the following properties hold

Property 3.

$$\forall t > t' > 0, \quad \psi^{\infty}(Y^{\infty}(-t), v_{-t}, \dots v_{-t'+1}) \preceq Y^{\infty}(-t'). \tag{15}$$

In other words, the trajectory of the direct chain for network $\mathcal{N}$ from time $-t$ to time 0, starting in $Y^{\infty}(-t)$ remains below the trajectory of the reversed chain from time 0 to time $-t$, starting in $Y^{\infty}(0)$.

Property 4. *For all $\mathbf{x}, \mathbf{t}, \mathbf{s}$, the trajectories $\psi(\mathbf{x}, \mathbf{v_{-t}}, \dots \mathbf{v_{-s+1}})$ are "correct" in the sense that if u_t are i.i.d. events with $\mathbb{P}(u_t = r_{ij}^{+/-}) = a_{ij}^{+/-}$, then $\psi(\mathbf{x}, \mathbf{v_{-t}}, \dots \mathbf{v_{-t'+1}})$ and $\psi(\mathbf{x}, \mathbf{u_1}, \dots \mathbf{u_s})$ have the same distribution.*

The proofs of the two properties are similar to the case without negative customers.

Algorithm 3: Envelope PSA with a Bounding Process (EPSA-BP)

```
Data:
(1) An infinite i.i.d. sequence $\{U_{-n}\}_{n\in\mathbb{N}}$ with $U_{-k}$ uniformly distributed in $[0,1]$ (together
with the sequence of backward events $\{w_{-n}\}_{n\in\mathbb{N}}$).
(2) A state $Y^{\infty}$ generated according to the distribution $\pi^{\infty}$.
Result: A state sampled from the stationary distribution of $\mathcal{N}$
begin
    $t := 1$;
    repeat
        for $k = \lfloor t/2 \rfloor$ to $t-1$ do
            $Y^{\infty} := \phi^{\infty}(Y^{\infty}, w_{-k})$;
            Compute the event $v_{-k}$ using the coupling procedure $(C_2)$;
        $Z^{(1)} := Y^{\infty} \wedge \mathbf{C}$; $Z^{(2)} := 0$;
        for $i = t-1$ downto $0$ do
            $Z^{(1)} := \sup_{\mathbf{x}\in[Z^{(1)},Z^{(2)}]} \psi(\mathbf{x}, v_{-i})$;
            $Z^{(2)} := \inf_{\mathbf{x}\in[Z^{(1)},Z^{(2)}]} \psi(\mathbf{x}, v_{-i})$;
        $t := 2t$;
    until $Z^{(1)} = Z^{(2)}$ ;
    return $Z^{(1)}$;
end
```

Theorem 4. *Algorithm EPSA-BP terminates with probability 1, and the output of EPSA-BP is a state with the stationary distribution of network $\mathcal{N}$.*

Proof. The following theorem can be proved in a similar way as Theorem 1. The only difference of the proof is in the forward part that uses the envelope technique in [Busic et al., 2012b] and the fact that the bounding chain also bounds the upper envelope of the original chain: For any t, any sequence $(u_{-t}, \dots, u_{-1}) \in [0,1]^t$, and any $\mathbf{x} \in \mathcal{S}$, $\mathbf{y} \in \mathcal{S}^{\infty}$, the backward bounding trajectory remains above the upper forward envelope.

$$\begin{aligned}
\mathbf{x} \preceq \mathbf{y} \quad &\Rightarrow \quad \psi(\mathbf{x}, v_{-t}, \dots, v_{-1}) \preceq \phi^{\infty}(\mathbf{y}, w_{-1}, \dots, w_{-t}) \\
&\Rightarrow \quad \sup_{\{\mathbf{z}|\mathbf{z}\leqslant\mathbf{x}\}} (\psi(\mathbf{z}, v_{-t}, \dots, v_{-1})) \preceq \phi^{\infty}(\mathbf{y}, w_{-1}, \dots, w_{-t}).
\end{aligned}$$

As for termination, in general, the envelopes do not necessarily meet ($Z^{(1)} = Z^{(2)}$) so the algorithm may not terminate with probability 1. In our case however, envelopes will meet with probability one for network $\mathcal{N}$ as soon as it is stable. This follows easily from the stability assumption (14).

□ □

We illustrated the extension to non-monotone networks on a concrete example of negative customers. In general, this extension will be valid for any Markov chain $Z(t+1) = \psi(Z(t), R_{t+1})$ under the following assumptions:

- It is possible to find a *bounding process* $Y(t+1) = \zeta(Y(t), R_{t+1})$ such that for any t, any sequence $(r_{-t+1}, \ldots, r_0) \in [0,1]^t$, and any $\mathbf{x} \in \mathcal{S}$, $\mathbf{y} \in \mathcal{S}'$,

$$\mathbf{x} \preceq \mathbf{y} \;\Rightarrow\; \psi(\mathbf{x}, r_{-t+1}, \ldots, r_0) \preceq \zeta(\mathbf{y}, r_{-t+1}, \ldots, r_0),$$

  where $\mathcal{S}$ and $\mathcal{S}'$ denote the state spaces of the original and bounding process respectively.

- It is easy to compute the backward transition function (β^∞ in the algorithm) for the bounding process and the initial sample distributed according to its stationary distribution. This second condition can be relaxed: we only need to be able to sample from an upper bound of this distribution.

- The *envelopes* for the original process couple and are easy to compute. For negative customer events r_{ij}^-, $1 \leqslant i, j \leqslant M$, we have:

$$\sup_{\mathbf{x}\in[\mathbf{y},\mathbf{z}]} \psi(\mathbf{x}, r_{ij}^-) = \begin{cases} \mathbf{z} - \mathbf{e}_i - \mathbf{e}_j, & y_i > 0, z_j > 0, \\ \mathbf{z} - \mathbf{e}_i, & y_i = 0, z_i > 0, \\ \mathbf{z}, & z_i = 0, \end{cases}$$

  and

$$\inf_{\mathbf{x}\in[\mathbf{y},\mathbf{z}]} \psi(\mathbf{x}, r_{ij}^-) = \begin{cases} \mathbf{y} - \mathbf{e}_i - \mathbf{e}_j, & y_i > 0, y_j > 0, \\ \mathbf{y} - \mathbf{e}_i, & y_i > 0, y_j = 0, \\ \mathbf{y} - \mathbf{e}_j, & y_i = 0, z_i > 0, y_j > 0, \\ \mathbf{y}, & z_i = 0. \end{cases}$$

Again, the coupling time is bounded by Equation (8) and does not depend on the capacities of the queues, unlike in classical perfect samplers.

8 Conclusion

In this paper we have presented a new perfect sampling algorithm for Jackson queuing networks with finite and infinite capacities in queues. A complexity analysis of the algorithm shows that its expected sampling time does not depend on the capacities. This is a remarkable improvement over classical perfect samplers whose sampling time increases at least linearly in the capacities, or fails when some capacities are infinite. Actually our approach is quite general and should be usable for any Markov chain for which a bounding process with a computable stationary distribution can be constructed and coupled with the original chain. In particular we have shown how it applies in the case of queuing networks with negative clients.

Acknowledgements

This work was partially supported by the ANR Numerical Models Program (MARMOTE project).

A Appendix: Solution of the traffic equations

Let us consider the traffic equations for a general Jackson network:

$$\left\{\begin{array}{lll} \forall i \neq 0, \quad \lambda_i & = & \sum_{j=1}^{M} p_{ji}(\lambda_j \wedge \mu_j) \\ \lambda_0 & = & \mu_0 \end{array}\right. \tag{16}$$

Theorem 5. *The system* (16) *admits exactly one solution.*

Proof. Let h be the function $h(\mathbf{x}) := \mathbf{x}', x'_i = \sum\limits_{j\in\mathcal{J}} p_{ji}(x_j \wedge \mu_j)$. P only has non negative coefficients, so h is clearly monotone on $\mathbb{R}^{\mathcal{J}}$.
We know that P is stochastic, so we have

$$h(\mathbf{x})_i = \sum_{j\in\mathcal{J}} p_{ji}(x_j \wedge \mu_j) \leq \sum_{j\in\mathcal{J}} p_{ji} \max(\boldsymbol{\mu}) \leq \max(\boldsymbol{\mu}) \Rightarrow h(x) \leq \max \boldsymbol{\mu} \text{ for } \mathbf{x} \in \mathbb{R}^{\mathcal{J}}, i,$$

setting an upper bound the image of h. We can also notice that for all input, the output coefficients are positive linear combinations of the input coefficients and of positive values ($\boldsymbol{\mu}$): if the input is component wise non negative, then so is the output.

Hence the restriction of h to the complete lattice $[0, \max \boldsymbol{\mu}]^{\mathcal{J}}$ is a monotone function to the same lattice. We can apply the Knaster-Tarski theorem: its fix points form a non empty lattice.

The minimum $\mathbf{x}^-$, maximum $\mathbf{x}^+$ of this lattice, and their difference $\boldsymbol{\delta} \overset{\text{def}}{=} \mathbf{x}^+ - \mathbf{x}^-$, verify:

$$\delta_i = \sum_{j\in\mathcal{J}, x_j^+ \leq \mu_j} p_{ji}\delta_j + \sum_{j\in\mathcal{J}, x_j^- < \mu_j < x_j^+} p_{ji}(\mu_j - x_j^-) \leq \sum_{j\in\mathcal{J}} \delta_j p_{ji}, \quad \text{for all } i \neq 0.$$

As for queue 0, $\delta_0 = x_0^+ - x_0^- = \mu_0 - \mu_0 = 0$. Since P is stochastic regular, its maximum eigenvalue is 1 and the corresponding eigenspace is of dimension one, generated by an eigenvector without null coordinates.

So $\boldsymbol{\delta}$ can only be the vector 0 meaning that the lattice is reduced to a singleton.
Hence the solution of the semi linear system exists and is unique. □ □

A.1 Computation

One could iterate the function h until reaching the fixed point, but this method would not give any guarantee on the number of iterations. Instead, we propose to work on a pair $(\mathcal{L}, \mathbf{x})$, where $\mathcal{L}$ is a set of queues and x is the solution of the following system, both system and solution will be updated together with the modification of $\mathcal{L}$.

$$\left\{\begin{array}{lll} x_0 & = & \mu_0 \\ x_i & = & \sum\limits_{j\in\mathcal{L}} \mu_j p_{ji} + \sum\limits_{j\notin\mathcal{L}} (x_j \wedge \mu_j) p_{ji} \quad for\ i \neq 0 \end{array}\right. \tag{17}$$

Starting with $\mathcal{L}$ equal to the set of all queues: $\mathcal{L} = \mathcal{J}$, we repeat the following procedure as long as possible.

- Compute the vector $\mathbf{x}$, solution of (17).
- Find one queue i^* in $\mathcal{L}$ such that $i^* \in \mathcal{L}, x_{i^*} \leq \mu_{i^*}$.
- set the new set $\mathcal{L} = \mathcal{L}\backslash\{i^*\}$

Lemma 6. *The solutions* $\mathbf{x}$ *of the successive systems form a non-increasing sequence and are expressible as linear combinations of* $\boldsymbol{\mu}_{|\mathcal{L}}$.

Proof. This proof is in two parts: we first prove that the solutions $\mathbf{x}$ form a non-increasing sequence.

Consider the family of functions $h_{\mathcal{L}}$ similar to the function h from the proof of Lemma 5 :

$$h_{\mathcal{L}}(\mathbf{x}) \overset{\text{def}}{=} \sum_{j \notin \mathcal{L}} p_{ji}(x_j \wedge \mu_j) + \sum_{j \in \mathcal{L}} p_{ji}\mu_j.$$

Every function of this family is monotone and admits only one fixed point on the lattice $[0, \max \boldsymbol{\mu}]^{\mathcal{J}}$ for exactly the same reason as in the proof of theorem 5.
We also have immediately (using $\forall a, b, (a \wedge b) \leq a$)

$$\forall \mathcal{L}, \mathcal{L}',\ \mathcal{L} \subset \mathcal{L}' \Rightarrow h_{\mathcal{L}} \preceq h_{\mathcal{L}'}$$

Using the characterization of the minimum fixed point as the lower bound of the set where the output is lower or equal to the input, we obtain:

$$\forall \mathcal{L}, \mathcal{L}',\ \mathcal{L} \subset \mathcal{L}' \Rightarrow \forall \mathbf{x}, h_{\mathcal{L}} \preceq h_{\mathcal{L}'} \Rightarrow h_{\mathcal{L}}(\mathrm{FP}(h'_{\mathcal{L}})) \leq \mathrm{FP}(h'_{\mathcal{L}}) \Rightarrow \mathrm{FP}(h_{\mathcal{L}}) \leq \mathrm{FP}(h_{\mathcal{L}'})$$

Since the sequence $\mathcal{L}$ is decreasing, the fixed points $\mathbf{x}$ of $h_{\mathcal{L}}$ form a non increasing sequence.

As a corollary, for any integer n and for any $i \in \mathcal{J} \backslash \mathcal{L}^n$, let n_0 be the step when i was chosen. We have $x_i^n \leq x_i^{n_0} \leq \mu_i$. Every term $(x_i \wedge \mu_i), i \notin \mathcal{L}^n$ in the system generated by $\mathcal{L}^n$ takes the value x_i. Therefore, the system is equivalent to the new system:

$$\left\{ \begin{array}{rcl} \forall i \neq 0, \quad x_i & = & \sum_{j \in \mathcal{L}} \mu_j p_{ji} + \sum_{j \notin \mathcal{L}} x_j p_{ji} \\ x_0 & = & \mu_0 \end{array} \right. \tag{18}$$

We denote by $S(\mathcal{L}, \boldsymbol{\mu})$ this new system, by $\mathbf{x}^n$ the solution of $S(\mathcal{L} = \mathcal{L}^n, \boldsymbol{\mu})$ and by i_n, the element chosen before the $(n+1)^{\text{th}}$ step.
Using the sequence $(\mathcal{L}^n)_n$, we build a sequence of matrices $(D^{(n)})_{n \in \mathbb{N}}$ such that for all index n, for all vector of real $\boldsymbol{\mu}$, $\mathbf{x}$ is solution of $S(\mathcal{L}^n, \boldsymbol{\mu})$ if and only if $\mathbf{x} = \boldsymbol{\mu} D^{(n)}$ with the additional property of partial sub-stochasticity :

$$D^n \text{ is } \mathcal{L}^n\text{-sub-stochastic if } \forall n, \forall i \in \mathcal{J}, \sum_{j \in \mathcal{L}^n} d_{ij}^n \leq 1$$

We take $D^{(0)} = P$; indeed, the solution $\mathbf{x}$ of $S(\mathcal{L}^0 = \mathcal{J}, \boldsymbol{\mu})$ verifies $\mathbf{x} = \boldsymbol{\mu} P$, and P is stochastic.

Computation of $D^{(n+1)}$ from $D^{(n)}$: We want to use the vector $(\mu_1, \ldots, \mu_{i_n - 1}, x_{i_n}, \mu_{i_n + 1}, \ldots, \mu_N)$ where coordinate i_n in $\boldsymbol{\mu}$ is replaced by the variable x_{i_n}.

We obtain a new system of variables $\mathbf{x}$ and $\mathbf{m}$ and equations :

$$\left\{ \begin{array}{l} \mathbf{x} \text{ solution of } S(\mathcal{L}^n, \mathbf{m}) \\ \mathbf{m} = (\mu_1, \ldots, \mu_{i_n - 1}, x_{i_n}, \mu_{i_n + 1}, \ldots, \mu_N) \end{array} \right.$$

With an immediate substitution on $\mathbf{m}$, we can see that this system is exactly $S(\mathcal{L}^{n+1}, \boldsymbol{\mu})$. By definition of D^n, this system is also equivalent to

$$\left\{ \begin{array}{l} \mathbf{x} = \mathbf{m} D^{(n)} \\ \mathbf{m} = (\mu_1, \ldots, \mu_{i_n - 1}, x_{i_n}, \mu_{i_n + 1}, \ldots, \mu_N) \end{array} \right.$$

in which we can operate the same substitution, therefore the solution $\mathbf{x}^{n+1}$ of $S(\mathcal{L}^{n+1}, \boldsymbol{\mu})$ is also a solution of $S(\mathcal{L}^n, \mathbf{m})$, so that it must satisfy $\mathbf{x}^{n+1} = (\mu_1, \ldots, \mu_{i_n-1}, x_{i_n}^{n+1}, \mu_{i_n+1}, \ldots, \mu_N)D^n$.

We notice that $x_{i_n}^{n+1}$ is the only variable appearing on right side of this system, and satisfies:

$$x_{i_n}^{n+1} = \sum_{j \in \mathcal{J} \setminus \{i_n\}} d_{ji_n}^n \mu_j + d_{i_n i_n}^n x_{i_n}^{n+1}$$

Hence[4]

$$x_{i_n}^{n+1} = \sum_{j \in \mathcal{J} \setminus \{i_n\}} \frac{d_{ji_n}^n}{1 - d_{i_n i_n}^n} \mu_j.$$

Now, we substitute in every equation of the system the variable $x_{i_n}^{n+1}$ by this combination. We obtain exactly the system $S(\mathcal{L}^{n+1}, \boldsymbol{\mu})$ under the form of a linear combination $\mathbf{x}^{n+1} = \boldsymbol{\mu} D^{n+1}$.

To summarize, we obtain D^{n+1} from D^n by applying the following successive elementary operations:

$$\begin{array}{lll} d_{ji_n}^{n+1} := \frac{d_{ji_n}^n}{1-d_{i_n i_n}^n} & & \text{for all } j \neq i_n \\ d_{ij}^{n+1} = d_{ij}^n + \frac{d_{ii_n}^n d_{i_n j}^n}{1-d_{i_n i_n}} = d_{ij}^n + d_{ii_n}^{n+1} d_{i_n j}^n & & \text{for all } i, j \neq i_n \\ d_{i_n j}^{n+1} := 0 & & \text{for all } j \end{array} \tag{19}$$

At last, let us show that D^{n+1} is $\mathcal{L}$-sub-stochastic:
Consider row $i \neq i_n$ (the i_n^{th} row is null), Using the previous result and the fact that $D^{(n)}$ is $\mathcal{L}^n$-sub-stochastic:

$$\sum_{j \in \mathcal{L}^{n+1}} d_{ij}^{n+1} = \sum_{j \in \mathcal{L}^n} d_{ij}^n + d_{ii_n}^n \Big(\sum_{j \in \mathcal{L}^n} d_{i_n j}^{n+1} - 1\Big) \leq \sum_{j \in \mathcal{L}^n} d_{ij}^n \leq 1$$

Also, by using (19) and the fact that all coefficient of d_{ij}^n are nonnegative, all coefficients d_{ij}^{n+1} are clearly nonnegative as well. Therefore, $D^{(n+1)}$ is $\mathcal{L}^{n+1}$-sub-stochastic.

□ □

Lemma 7. *The generated sequence stops after at most $|\mathcal{J}|$ steps and can only end on a solution of the traffic equations.*

Proof. At each step, the size of $\mathcal{L}$ decreases by one. This size begins at $|\mathcal{J}|$ and cannot be negative. Hence the sequence cannot have more than $|\mathcal{J}|$ steps.

An ending is a state where every element of $\mathcal{L}$ satisfies: $\mathbf{x}_i^{\text{end}} > \mu_i$ for $i \in \mathcal{L}$ (stopping condition) and $\mathbf{x}_i^{\text{end}} \leq \mu_i$ for $i \in \mathcal{J} \setminus \mathcal{L}$ (part of recurrence property).

Hence the min operator $\wedge$ in (7) gives μ_i on exactly the set $\mathcal{L}$. Using function h defined in the proof of theorem 5 (on page 28), we have

$$\forall i \in \mathcal{J},\, h(\mathbf{x}^{\text{end}})_i = \sum_{j \in \mathcal{L}} p_{ji} \mu_i + \sum_{j \notin \mathcal{L}} p_{ji} \mathbf{x}_i^{\text{end}} = \mathbf{x}_i^{\text{end}}$$

by definition of $\mathbf{x}^{\text{end}}$ as the solution of exactly this system. Since $\mathbf{x}^{\text{end}}$ is a fixed point of h, it is also the only solution of the system (7). □

[4] We could prove that if the initial network was well formed, no diagonal coefficient can become 1. Instead, we can show that *even* this case is not problematic. Let us suppose that $d_{i_n i_n} = 1$. We know that D^n is sub stochastic, so every other coefficient in the line is null, so the incorrect value is not used.

Algorithm 4: solving the system

```
Data: $P$ a stochastic matrix, $\boldsymbol{\mu}$ a real vector
Result: $\boldsymbol{\lambda}^{in}$ the solution of the system, $\mathcal{G}$ the set of unstable queues
begin
    $D \leftarrow P$ ;
    $\forall i \neq 0,\ x_i \leftarrow \sum_{j \in \mathcal{J}} \mu_j p_{ji}$ and $x_0 \leftarrow \mu_0$ ;
    $\mathcal{L} \leftarrow \mathcal{J} - \{0\}$ ;
    while $\{i \in \mathcal{L}, x_i \leq \mu_i\} \neq \emptyset$ do
        choose $i^* \in \{i \in \mathcal{L}, x_i \leq \mu_i\}$ ;
        $k \leftarrow \frac{1}{1-d_{i^*i^*}}$ ;
        $\forall i \in \mathcal{J}, j \in \mathcal{L}, d_{ij} \leftarrow d_{ij} + k \cdot d_{ii^*} \cdot d_{i^*j}$ ;
        $\mathcal{L} \leftarrow \mathcal{L} \backslash i^*$;
        $\forall i \in \mathcal{L},\ x_i \leftarrow \sum_{j \in \mathcal{L}} \mu_j d_{ji}$ ;
    $\forall i \in \mathcal{J} - \{0\},\ x_i \leftarrow \sum_{j \in \mathcal{L}} \mu_j d_{ji}$ ;
    return $(\mathbf{x}, \mathcal{L})$
end
```

Remark 4. *We don't need to implement the instruction* $\{\forall j \in \mathcal{J}, d_{ji^*} \leftarrow 0\}$ *since the coefficients in column outside* $\mathcal{L}$ *will never be accessed again.*

Proposition 3 (complexity)**.** *Algorithm 4 computes the solution of the traffic equations in* $O(M^3)$ *operations.*

Proof. There will be at most M executions of the loop.
Cost of the different operations :

1. **initialization**: The first step is a product of a matrix and a vector: $O(M^2)$
2. **choice of** i_n: At most $|\mathcal{L}|$ tests are done in one set: $O(M)$.
3. **update of** D: At each step less than M^2 coefficients are updated, and each update consists in a constant number of operations.
4. **update of** $\mathcal{L}$: With a direct pointer, this operation is done in constant time.
5. **computation of x** It is a matrix-vector product : done in $O(M^2)$

Finally, the algorithm is made of at most M repetitions of this $O(M^2)$ loop, so its execution is in $O(M^3)$. □ □

References

Anselmi, J. and Gaujal, B. (2011). On the efficiency of perfect simulation in monotone queueing networks. In *IFIP Performance: 29th International Symposium on Computer Performance, Modeling, Measurements and Evaluation*, Amsterdam. ACM Performance Evaluation Review.

Anselmi, J. and Gaujal, B. (2013). Efficiency of simulation in monotone hyper-stable queueing networks. *Queuing Systems, Theory and Applications.*

Bolch, G., Greiner, S., de Meer, H., and Trivedi, K. (2005). *Queueing Networks and Markov Chains*. Wiley-Interscience.

Busic, A., Gaujal, B., and Perronnin, F. (2012a). Perfect Sampling of Networks with Finite and Infinite Capacity Queues. In Al-Begain, K., Fiems, D., and Vincent, J.-M., editors, *19th International Conference on Analytical and Stochastic Modelling Techniques and Applications (ASMTA) 2012*, volume 7314 of *Lecture Notes in Computer Science*, pages 136–149, Grenoble, France. Springer.

Busic, A., Gaujal, B., and Pin, F. (2012b). Perfect sampling of Markov chains with piecewise homogeneous events. *Performance Evaluation*, 69(6):247 – 266.

Busic, A., Gaujal, B., and Vincent, J.-M. (2008). Perfect simulation and non-monotone Markovian systems. In *3rd International Conference Valuetools'08*, Athens, Greece. ICST.

Chen, H. and Yao, D. D. (2001). *Fundamentals of Queueing Networks*. Springer-Verlag.

Dopper, J., Gaujal, B., and Vincent, J.-M. (2006). Bounds for the coupling time in queueing networks perfect simulation. In *Celebration of the 100th anniversary of Markov*, pages 117–136.

Gaujal, B., Perronnin, F., and Bertin, R. (2008). Perfect simulation of a class of stochastic hybrid systems with an application to peer to peer systems. *Journal of Discrete Event Dynamic Systems*, 18(2):211–240. Special Issue on Hybrid Systems.

Gelenbe, E. (1991). Product-form queueing networks with negative and positive customers. *Journal of Applied Probability*, 28(3):pp. 656–663.

Goodman, J. B. and Massey, W. A. (1984). The non-ergodic jackson network. *Journal of Applied Probability*, 21(4):860–869.

Jackson, J. R. (1963). Job shop-like queueing systems. *Management Sci.*, 10,131.

Kelly, F. (1979). *Reversibility and Stochastic Networks*. John Wiley & Sons Ltd.

Kendall, W. S. (2005). Notes on perfect simulation. Dept. of statistics, University of Warwick.

Kendall, W. S. and Møller, J. (2000). Perfect simulation using dominating processes on ordered spaces, with application to locally stable point processes. *Advances in Applied Probability*, 32(3):844–865.

Pin, F., Busic, A., and Gaujal, B. (2011). Acceleration of perfect sampling for skipping events. In *Valuetools*, Paris.

Propp, J. G. and Wilson, D. B. (1996). Exact sampling with coupled Markov chains and applications to statistical mechanics. *Rand. Struct. Alg.*, 9(1-2):223–252.

Vincent, J.-M. (2005). Perfect simulation of monotone systems for rare event probability estimation. In *WSC '05: Proceedings of the 37th conference on Winter simulation*, pages 528–537. Winter Simulation Conference.

Walker, A. J. (1977). An efficient method for generating discrete random variables with general distributions. *ACM Trans. Math. Softw.*, 3:253–256.

RESEARCH CENTRE
GRENOBLE – RHÔNE-ALPES

Inovallée
655 avenue de l'Europe Montbonnot
38334 Saint Ismier Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399