Perfect Sampling with Aggregated Envelopes

Ana Bušić
INRIA and ENS
Paris, France
Email: ana.busic@ens.fr

Emilie Coupechoux
INRIA and ENS
Paris, France
Email: emilie.coupechoux@ens.fr

***Abstract*—Perfect sampling is a technique that uses coupling arguments to provide a sample from the stationary distribution of a Markov chain. This technique is efficient if the transition function of the Markov chain is monotone. In the non-monotone case, one can construct bounding chains that can detect whether the initial chain has coupled. For instance, if the state space is a lattice, then one such bounding chain can be defined by taking the smallest interval that contains the image of the one step transition function. Here we propose to combine the ideas of bounding processes and the aggregation of Markov chains.**

We illustrate the proposed approach of aggregated envelope bounding chains on the service tools model proposed by Vliegen and Van Houtum (2009). For this model, the aggregated envelope method allows to reduce exponentially the dimension of the state space and allows effective perfect sampling algorithms. Under some conditions on the transition rates (high service case), the running time of our algorithm is linear in terms of the total capacity in the system.

***Index Terms*—Perfect sampling, Markov chains, queueing systems.**

I. Introduction

A Perfect Sampling Algorithm (PSA) for finite Markov chains has been introduced by Propp & Wilson [1] using a *coupling from the past* scheme. Perfect sampling procedures have been developed since in various contexts. We mention here only some works directly linked to the present article. For more information, see the annotated bibliography: *Perfectly Random Sampling with Markov Chains*, http://dimacs.rutgers.edu/~dbwilson/exact.html/.

The main drawback of the initial Propp & Wilson PSA is the need to consider the coupled trajectories from *all possible initial states* of the Markov chain. This can be avoided if the transition function of the Markov chain is monotone, as in that case one can consider only trajectories from *extremal* initial states (two if the state space is a finite lattice) [1]. Similarly, a perfect sampling algorithm for chains with anti-monotone events was given in [2]. These techniques have been successfully used in [3] to construct a PSA for networks of queues.

In the case of general non-monotone chains, Kendall and Møller [4] proposed to consider a new bounding process, that sandwiches all the original trajectories of the Markov chain. This bounding process is usually derived ad-hoc for a given Markov chain with specific simplifying properties. Bušić et al. [5] proposed Envelope Perfect Sampling Algorithm (EPSA), that constructs a bounding Markov chain in the case when the state space is a finite lattice. Intuitively, the bounding chain used by EPSA is defined on the non-empty intervals of the state space, and the transition function of this envelope chain is obtained by taking the smallest interval that contains the image of the one step transition function of the original chain. We give a more formal overview of EPSA in Section II.

The clear advantage of EPSA is the fact that one needs to consider only one trajectory of the envelope chain (the one that starts with the initial interval equal to the whole state space). However, the computation of the envelope transition function can be very difficult. In some extreme cases, it can be linear in the cardinality of the state space, so that there is no gain compared to the original PSA. In many cases of queueing networks, however, the complexity of the envelope transition function is linear in the dimension of the state space [6]. For many applications, this is more than acceptable and allows effective perfect sampling algorithms.

However, in many variants of loss networks (see Kelly [7]), the dimension of the state space is exponential with the number of different resources in the system. In loss networks, demands, for example phone calls, need several links to be simultaneously available. If all links are available, the call is connected. After the call is finished, all links are simultaneously released. When one or more of the links are not available, the call does not connect, and the demand for all links is lost. Although loss networks have a product-form solution, exactly computing the blocking probabilities for this system is known to be a difficult problem (Louth et al. [8]), due to the normalizing constant.

The main issue we address in this paper is how to handle the state space explosion problem in perfect sampling algorithms. We propose a new method of aggregated envelopes, that combines EPSA with the aggregation of Markov chains. For loss networks and its variants, our aggregated envelope method allows to reduce exponentially the dimension of the state space and allows effective perfect sampling algorithms. Under some conditions on the transition rates (high service rate), the running time of our algorithm is linear in terms of the total capacity in the system.

Our work is motivated by a variant of loss networks, called the service tools model, introduced by Vliegen and van Houtum [9]. In their problem, to perform a maintenance action, several service tools are needed at the same time. Whenever one or more tools are not present, they are sent by an emergency shipment to enable the initiation of the maintenance action as soon as possible. For the supply location

under consideration the demand for these emergency shipped tools is lost. Furthermore, after usage all tools return to the location they were sent from together. The difference between the service tools model and loss networks is only at the boundary of the state space: If some of demanded resources are not available, in the service tools model the available resources are allocated, while in loss networks the entire demand is lost. This difference breaks the product form and makes the analysis considerably more difficult. Vliegen and van Houtum [9] developed different approximations. Bušić et al. [10] proved that some of the previous approximations give bounds for the original system. However, for larger instances those bounds are still time consuming, and there is in general no guarantee on their accuracy.

Both loss networks and the service tools models can be seen as special cases of assemble-to-order queueing systems (ATO), whose general characteristics are that both arrivals and services in different queues can occur simultaneously. For an overview on ATO systems, see Song and Zipkin [11]. In the framework of ATO systems, we focus here on *continuous-review models*, with *exponential replenishment times* and *finite stock capacities*. We consider two different options for the out-of-stock situation: A demand can be fulfilled partly (as in the service tools case), referred to as *partial order service* (POS); or lost fully (as in loss networks), referred to as *total order service* (TOS). Also, we can distinguish between two different situations for the service/return of components: components are either returned *individually* (and independently) or *jointly* (as in loss networks and its variants).

In the case of ATO systems with individual service, Song et al. [12] proposed an exact evaluation, both for TOS and POS, by using a matrix geometric approach. This exact method, however, is computationally inefficient for larger problem instances. Dayanik et al. [13] presented several approximations and bounds on the performance of ATO-POS systems with individual returns. We can conclude that all four cases (POS/TOS with individual/joint services) of ATO systems can be difficult to analyze directly. Simulation can thus be an interesting alternative for bounding or approximation techniques developed in the literature.

In the case of individual returns, the state space is an I-dimensional lattice (where I is the number of different item types), so EPSA can be used directly. For the case of joint services, loss networks have a product-form solution, which makes the exact calculation somewhat easier. We therefore focus on the case of the ATO-POS system with joint services, i.e. the service tools model in [9]. However, we would like to highlight that the presented approach can be very easily adapted for loss networks, where it can be used to estimate the normalizing constant of the product form solution.

In Section II we give an overview of perfect sampling technique and the related literature. In Section III we introduce the method of aggregated envelopes. In Sections IV and V we illustrate this method on the example of the service tools model. Finally, in Section VI we discuss some possible extensions of our work and provide conclusions.

II. Perfect sampling and the method of envelopes

The evolution of a finite Discrete Time Markov Chain (DTMC) can always be obtained using a finite number of discrete events (or actions). We consider a system description similar to Generalized Semi Markov Processes [14], with a focus on state changes rather than on time: we consider the tuple $\mathcal{M} = (\mathcal{X}, \mathcal{E}, \nu, f)$ where $\mathcal{X}$ is a finite state space, $\mathcal{E}$ is the *set of events*, ν is a probability distribution on $\mathcal{E}$, and f is a *transition function*, $f : \mathcal{X} \times \mathcal{E} \to \mathcal{X}$.

This transition function f can be naturally extended to words $a_{1\to t} \stackrel{\text{def}}{=} a_1 a_2 \dots a_t \in \mathcal{E}^t$, $t \in \mathbb{N}$ (where $a_{1\to 0} := \epsilon$ is the empty word). For any $t \in \mathbb{N}$, $f : \mathcal{X} \times \mathcal{E}^t \to \mathcal{X}$ is defined by: $f(x, \epsilon) \stackrel{\text{def}}{=} x$ and $f(x, a_{1\to t}) \stackrel{\text{def}}{=} f(\dots f(f(x, a_1), a_2), \dots, a_t)$.

Let $(a_t)_{t\geq 1} = (a_1, \dots, a_t, \dots)$ be an infinite *i.i.d.* sequence of random events in $\mathcal{E}$, distributed according to ν. Then for any $x_0 \in \mathcal{X}$, the random process $(X_t \stackrel{\text{def}}{=} f(x_0, a_{1\to t}))_{t\geq 0}$ is a Markov chain issued from x_0 with probability transition matrix P given by:

$$\text{for all } x, y \text{ in } \mathcal{X}, \quad P(x,y) = \sum_{a\in\mathcal{E},\ f(x,a)=y} \nu(a). \qquad (1)$$

We say that the Markov chain (X_t) is *generated* by $\mathcal{M}$ and $(a_t)_{t\geq 1}$.

Conversely, for any probability transition matrix P on a finite state space $\mathcal{X}$, it is easy to see that there exists a tuple $\mathcal{M} = (\mathcal{X}, \mathcal{E}, \nu, f)$ such that (1) holds, *i.e.* such that $\mathcal{M}$ generates a Markov chain on $\mathcal{X}$ with transition matrix P, but that representation is in general not unique. However, such a representation naturally arises for many systems, including Markovian queueing networks.

We can build a family of Markov chains $\{(X_t(x) = f(x, a_{1\to t}))_{t\geq 0} \mid x \in \mathcal{X}\}$ starting from each state $x \in \mathcal{X}$, referred to as the *grand coupling* generated by $\mathcal{M}$ and $(a_t)_{t\geq 1}$ [15]. We will say that the grand coupling has *coupled* (or more precise *coalesced*) at time t if all the Markov chains of the family has reached the same state. Using the notation $f(U, a_{1\to t}) \stackrel{\text{def}}{=} \{f(x, a_{1\to t}), x \in U\}$ for any subset $U \subset \mathcal{X}$, this is equivalent to the fact that $f(\mathcal{X}, a_{1\to t})$ is reduced to a singleton. In the following, $|V|$ denotes the cardinality of set V.

A. Perfect Sampling

Let $(X_t)_{t\in\mathbb{N}}$ be an irreducible and aperiodic DTMC with finite state space $\mathcal{X}$ and transition matrix P. Consider a discrete event system representation $\mathcal{M} = (\mathcal{X}, \mathcal{E}, \nu, f)$ that satisfies (1), and let π denote the steady state distribution of the chain: $\pi = \pi P$. Perfect Sampling Algorithm (PSA) allows one to draw a steady state distributed random variable in finite time, using coupling from the past.

Theorem 1 (Propp and Wilson [1])**.** *Let* $(a_{-t})_{t\in\mathbb{N}} = (a_0, a_{-1}, \dots, a_{-t}, \dots)$ *be a sequence of events* i.i.d. *with distribution* ν *on* $\mathcal{E}$*. There exists* $\ell \in \mathbb{N}$ *such that* $\lim_{t\to\infty} \left|f(\mathcal{X}, a_{-t+1\to 0})\right| = \ell$ *almost surely. The grand coupling generated by* $\mathcal{M}$ *and* $(a_{-t})_{t\in\mathbb{N}}$ *is coalescing if*

$\ell = 1$. *In that case, let* $\tau \stackrel{\text{def}}{=} \inf\left\{t : \left|f(\mathcal{X}, a_{-t+1\to 0})\right| = 1\right\}$ *be the coupling time of the chain. Then* $\mathbb{E}(\tau) < \infty$ *and* $f(\mathcal{X}, a_{-\tau+1\to 0})$ *is steady state distributed.*

The main drawback of PSA is the fact that one needs to simulate one Markov chain starting from each state in $\mathcal{X}$, that could be too large for a practical use of the algorithm. Several approaches have been used to overcome this problem. The main one for a partially ordered state space $(\mathcal{X}, \preceq)$ and monotone events was already given in [1].

Definition 2. *An event* $a \in \mathcal{E}$ *is said to be monotone if, for all* $x, y \in \mathcal{X}$, $x \preceq y \Rightarrow f(x,a) \preceq f(y,a)$.

If all events are monotone, then one can consider only the trajectories issued from maximal and minimal initial states [1].

In the case of general non-monotone chains, it is possible to use a bounding chain method, introduced by Kendall and Møller in [4]. EPSA (Envelope Perfect Sampling Algorithm) [5] constructs bounding chains in the case when the state space is equipped with a lattice order relation. We give next a short overview of EPSA.

B. Bounding Interval Chains

We assume that the state space $(\mathcal{X}, \preceq)$ is a lattice. For $m, M \in \mathcal{X}$, denote by $[m, M] \stackrel{\text{def}}{=} \{x \in \mathcal{X} : m \preceq x \preceq M\}$ the *(lattice) interval* between the endpoints m and M (note that $[m, M] \neq \emptyset$ if and only if $m \preceq M$). Let $\mathcal{J}$ be the set of all nonempty lattice intervals: $\mathcal{J} = \{[m, M] : m, M \in \mathcal{X},\ m \preceq M\}$. Given a grand coupling $\{(X_t(x))_{t\geq 0} \mid x \in \mathcal{X}\}$, a *bounding interval chain* is any Markov chain of nonempty intervals $([m_t, M_t])_{t\geq 0}$ such that: for all x in $\mathcal{X}$ and all $t \geq 0$, $X_t(x) \in [m_t, M_t]$. In particular we notice that when $m_t = M_t$, the grand coupling has necessarily coalesced.

A new *envelope transition* function $F : \mathcal{J} \times \mathcal{E} \to \mathcal{J}$, that transforms intervals into intervals, is defined by: for all $[m, M] \in \mathcal{J}$ and $a \in \mathcal{E}$,

$$F([m, M], a) \stackrel{\text{def}}{=} \left[\inf_{m\preceq x\preceq M} f(x,a),\ \sup_{m\preceq x\preceq M} f(x,a)\right].$$

As with f, transition function F can be extended to a finite word of events $a_{1\to t} = a_1 \dots a_t \in \mathcal{E}^t$, $t \in \mathbb{N}$: $F([m, M], a_{1\to t}) \stackrel{\text{def}}{=} F\left(\dots F\left(F\left([m, M], a_1\right), a_2\right), \dots, a_t\right)$. Let $\bot \stackrel{\text{def}}{=} \inf \mathcal{X}$ (resp . $\top \stackrel{\text{def}}{=} \sup \mathcal{X}$) be the bottom (resp. top) element of $\mathcal{X}$. The process $[m_t, M_t] \stackrel{\text{def}}{=} F([\bot, \top], a_{1\to t})$ is a Markov chain over the state space $\mathcal{X} \times \mathcal{X}$, called the *envelope chain*, and is a bounding interval chain of the grand coupling $\{(f(x, a_{1\to t}))_{t\geq 0} \mid x \in \mathcal{X}\}$. Notice that the lower envelope $(m_t)_{t\in\mathbb{N}}$ alone is not a Markov chain, neither is the upper one $(M_t)_{t\in\mathbb{N}}$, since they depend on each other.

The envelope process can be used to detect the coalescence of the grand coupling. The following result was shown in [5]:

Theorem 3. *Let* $(a_{-t})_{t\in\mathbb{N}}$ *be a sequence of events* i.i.d. *with distribution* ν *on* $\mathcal{E}$. *Assume that the envelope chain* $F([\bot, \top], a_{-t+1\to 0})$ *hits the set of single point intervals* $\mathcal{P} = \{[x, x] : x \in \mathcal{X}\}$ *a.s. in finite time. Let* $\tau_e \stackrel{\text{def}}{=} \min\{t : F([\bot, \top], a_{-t+1\to 0}) \in \mathcal{P}\}$, *then* τ_e *is a backward coupling time of the envelope chain. The state defined by* $F([\bot, \top], a_{-\tau_e+1\to 0})$ *has the steady state distribution of DTMC* $(X_t)_{t\in\mathbb{N}}$.

Algorithm 1: EPSA [5]

Data: *I.i.d.* events $(a_{-t})_{t\in\mathbb{N}} \in \mathcal{E}^{\mathbb{N}}$
Result: A state $x^* \in \mathcal{X}$ generated according to the stationary distribution of the Markov chain
begin
  $t := 1$;
  repeat
    $m := \bot$; $M := \top$;
    for $i = t - 1$ **downto** 0 **do**
      $[m, M] := F\left([m, M], a_{-i}\right)$;
    $t := 2t$;
  until $m = M$;
  $x^* := m$;
  return x^*;
end

Envelope Perfect Sampling Algorithm (EPSA) is given in Algorithm 1. The reason to double t at each loop of the algorithm is that we need to compute $F([\bot, \top], a_{-t+1\to 0})$ in each loop, which corresponds to t iterations of F. While increasing t by 1 at each loop would lead to a quadratic cost in τ_e, doubling it keeps the complexity linear. This was already observed in [1].

III. Aggregated envelopes

In many applications, the main drawback of the perfect sampling method is the cardinality of the state space. In the case of ATO systems with joint services (loss networks and its variants), the dimension of the state space is exponential with respect to the number of different resource types. Indeed, although we are usually interested only in the total number of available resources of each type, this information is not sufficient to describe the evolution of the system: In order to have a Markov chain, we need to track detailed information on which resources were allocated together (as they will be released together). Thus even storing the vector representing the state of the system becomes challenging.

The idea of aggregated envelope method is to consider only the projection of the state space on a smaller space. However, this projection does not contain all the information about the evolution of the system so we will need to construct a bounding chain that takes into account all such possible evolutions.

We assume that our initial Markov chain is given by a tuple $\mathcal{M} = (\mathcal{N}, \mathcal{E}, \nu, g)$. We assume further that there is a projection function $\psi : \mathcal{N} \to \mathcal{X}$ such that $(\mathcal{X}, \preceq)$ is a finite lattice. The state space $\mathcal{N}$ is not necessarily a lattice (we do not assume any ordering relation on $\mathcal{N}$). In fact, this is another important motivation for the aggregated envelope method.

In practice, $|\mathcal{X}|$ will be much smaller that $|\mathcal{N}|$ (in the model considered in Sections IV and V, the dimension of space $\mathcal{N}$

is $2^I - 1$, where I is the dimension of space $\mathcal{X}$).

In this section, we develop a method that allows to sample an interval of $\mathcal{X}$ containing the projection of a state distributed according to the stationary distribution π of the original Markov chain. We will see that, sometimes (e.g. ATO systems with joint services under high service rate assumption, see Section V), it is even possible to sample a state in $\mathcal{N}$, distributed according to the stationary distribution.

A. The idea of aggregation

Our starting idea is to use the projected state space $\mathcal{X}$ for simulations. Intuitively, the original Markov chain $X = (X_t)_{t\in\mathbb{N}}$ evolves in $\mathcal{N}$, but we can only observe its projection $Y = (Y_t)_{t\in\mathbb{N}} = (\psi(X_t))_{t\in\mathbb{N}}$. Assume that the original chain is in state $n_0 \in \mathcal{N}$. The only information we have is $x = \psi(n_0) \in \mathcal{X}$. When an event $a \in \mathcal{E}$ occurs, we need to determine the next state in $\mathcal{X}$. As Y is not a Markov chain, we cannot determine the next state only from knowing x. Instead, we will consider the evolution from *all* the states $n \in \mathcal{N}$ such that $\psi(n) = x$. More formally, for $x \in \mathcal{X}$, we consider the following *subset* $S_x \subset \mathcal{N}$:

$$S_x = \{n \in \mathcal{N}, \psi(n) = x\} = \psi^{-1}(\{x\}).$$

Let $\mathcal{P}(\mathcal{X})$ denotes the family of subsets of $\mathcal{X}$, and $f : \mathcal{P}(\mathcal{X}) \times \mathcal{E} \to \mathcal{P}(\mathcal{X})$ a transition function defined by:

$$f(U, a) = \cup_{x\in U}\{\psi\left(g(S_x, a)\right)\},$$

where g is the transition function of the original Markov chain.

If the projected space is small enough, we can use Algorithm 2.

Algorithm 2: Perfect Sampling using Aggregation

Data: *I.i.d.* events $(a_{-t})_{t\in\mathbb{N}} \in \mathcal{E}^{\mathbb{N}}$.
Result: Subset $U \subset \mathcal{X}$ containing the projection of a state $n^* \in \mathcal{N}$ distributed according to π.

```
begin
    t = 1; c = 0;
    repeat
        U = X;
        for i = t − 1 downto 0 do
            U := f(U, a_{−i}) ;
            if |U| = 1 then c = 1;
        t := 2t;
    until c = 1 ;
    return U;
end
```

The lack of knowledge induced by the projection on $\mathcal{X}$ forces to consider *all* the states $n \in \mathcal{N}$ with the same projection x. This induces two main problems:

- Even if the original system couples, we may never have $|U| = 1$ in Algorithm 2.
- Even if at some time $-t$ we have only one value for the projected process (i.e. $|U| = 1$), this is not necessarily the case for times $-s$, $-t \le -s \le 0$, as in fact this single projected value at time $-t$ can correspond to many different states in $\mathcal{N}$.

In addition to these problems, space $\mathcal{X}$ can be too big to consider all the initial states $x \in \mathcal{X}$. So, first, we would like to be able to compute $\psi\left(g(S_x, a)\right)$, for a given state x and event a, without considering all $n \in S_x$ (for ATO systems with joint services, $|S_x|$ grows exponentially with the number of item types). Also, we do not want to be forced to calculate, at each step, $\psi\left(g(S_x, a)\right)$ for *all* the current states x (even if we are able to calculate it easily for each state). To overcome this, we will combine the idea of aggregation with the method of envelopes developed in [5].

B. Combining aggregation and envelopes

First we need to define the Markov chain $Y^{\inf}$ (respectively $Y^{\sup}$) that maps x to the infimum (resp. supremum) of $f(\{x\}, a) = \psi\left(g(S_x, a)\right)$. We consider the following transition functions: for all $x \in \mathcal{X}$ and $a \in \mathcal{E}$,

$$g^{\inf}(x, a) \stackrel{\text{def}}{=} \inf f(\{x\}, a), \quad g^{\sup}(x, a) \stackrel{\text{def}}{=} \sup f(\{x\}, a).$$

Let $m, M \in \mathcal{X}$ such that $m \preceq M$. The method of envelopes changes the subset $[m, M]$ into a new subset $[m', M']$ (that depends on the event $a \in \mathcal{E}$ and that usually involves only the transition function of the Markov chain we consider). Here we are considering two Markov chains $Y^{\inf}$ and $Y^{\sup}$ on the same space $\mathcal{X}$, with the same set of events $\mathcal{E}$, but with two different transition functions $g^{\inf}$ and $g^{\sup}$. Considering separately the envelopes of the infimum and the supremum chains does not necessarily sandwich the projected process (if either $g^{\inf}$ or $g^{\sup}$ is not monotone). We define the aggregated envelope transition function as follows: For $m, M \in \mathcal{X}$ such that $m \preceq M$, and $a \in \mathcal{E}$,

$$\begin{aligned} H\left([m, M], a\right) &\stackrel{\text{def}}{=} \left[\inf_{m\preceq x\preceq M} g^{\inf}(x, a), \sup_{m\preceq x\preceq M} g^{\sup}(x, a)\right] \\ &= \left[\underline{H}\left([m, M], a\right), \overline{H}\left([m, M], a\right)\right]. \end{aligned}$$

In order to compare the projected process $Y = \psi(X)$ of the original chain X to the lower and the upper envelopes of H, we need the following notations. Let Z be a process with state space $\mathcal{Z}$. For $(t, z) \in \mathbb{N} \times \mathcal{Z}$, the notation $Z(-t, z)$ stands for a realization of Z that starts from z at time $-t$, while $Z_{-s}(-t, z)$ denotes the value of this realization at time $-s$. The next lemma shows that the chain with transition function H is a bounding interval chain for the projected process $\psi(X)$.

Lemma 4. *Let $n \in \mathcal{N}$ and $y, z \in \mathcal{X}$ such that $y \preceq \psi(n) \preceq z$. Let $t \in \mathbb{N}$ and $a_{-t+1}, \ldots, a_0 \in \mathcal{E}$. Then we have for any $s \le t$:*

$$\begin{aligned} \underline{H}\left([y, z], a_{-t+1\to -s}\right) &\preceq \psi(X_{-s}(-t, n)) \\ &\preceq \overline{H}\left([y, z], a_{-t+1\to -s}\right). \end{aligned}$$

Proof: Let $x := \psi(n)$. We prove the result by descending induction on s. First, for $s = t$ the result is trivial, as $a_{-t+1\to -s} = \epsilon$ (empty word), so

$$\underline{H}\left([y, z], \epsilon\right) = y \preceq \psi(X_{-t}(-t, n)) \preceq z = \overline{H}\left([y, z], \epsilon\right).$$

Assume now the result is true for some s, $s \le t$. Let $n' := X_{-s}(-t, n)$, $x' := \psi(n')$, $y' := \underline{H}\left([y, z], a_{-t+1\to -s}\right)$ and

$z' := \overline{H}([y,z], a_{-t+1\to -s})$. Then we have that $y' \preceq x' \preceq z'$ by induction hypothesis. For $s-1$, we have:

$$\begin{aligned}\underline{H}([y',z'], a_{-s+1}) &\preceq g^{\inf}(x', a_{-s+1}) \preceq \psi(X_{-s+1}(-t,n)) \\ &\preceq g^{\sup}(x', a_{-s+1}) \preceq \overline{H}([y',z'], a_{-s+1}).\end{aligned}$$

By the definition of y' and z', $\underline{H}([y',z'], a_{-s+1}) = \underline{H}([y,z], a_{-t+1\to -s+1})$ and $\overline{H}([y',z'], a_{-s+1}) = \overline{H}([y,z], a_{-t+1\to -s+1})$, which gives the result for $s-1$. ■

The aggregated envelope method is summarized as Algorithm 3.

Algorithm 3: Aggregated Envelope Perfect Sampling

Data: *I.i.d.* events $(a_{-t})_{t\in\mathbb{N}} \in \mathcal{E}^{\mathbb{N}}$.
Result: Interval $[m^*, M^*] \subset \mathcal{X}$ containing the projection of a state $n^* \in \mathcal{N}$ distributed according to π.

```
begin
    t = 1; c = 0;
    repeat
        m := ⊥ (∈ X); M := ⊤ (∈ X);
        for i = t − 1 downto 0 do
            [m, M] := H([m, M], a_{−i}) ;
            if m = M then c = 1;
        t := 2t;
    until c = 1 ;
    m* := m; M* := M;
    return m*, M*;
end
```

IV. ATO-POS WITH JOINT SERVICES

A. Model description

We consider an assemble-to-order system with partial order service (ATO-POS) and joint returns of items, also called the service tools model by Vliegen and van Houtum [9]. We assume that there are I different item types and denote by $\mathcal{I} = \{1, \dots, I\}$. For each $i \in \mathcal{I}$, let C_i be the total amount of items of type i (i.e. we consider finite stock capacities). The customers arrive in the system according to a Poisson process of rate λ. Each customer asks for a subset of items and the probability to ask subset A is denoted by p_A. Thus the demands for each subset A follows a Poisson process of rate $\lambda_A = p_A\lambda$. If some demanded items are not available, then the customer takes the available items (POS case) and these items return from the customer together (joint service) after an exponential time of rate μ. The demand for the items that are not available is lost.

This system can be modeled as a network of I queues with joint arrivals and services: arrivals to queues represent demands for different subsets of items and service in a queue models returns of items. Denote by $C = (C_1, \dots, C_I)$ the vector of queue capacities.

We consider joint services, so that items that entered the system together (borrowed by one customer) will also leave the system together (returned from the customer). Therefore we need to keep memory of the way items entered the queues together. The system can be modeled as a continuous time Markov chain with state space:

$$\mathcal{N} = \left\{ (n_A)_{A\subseteq\mathcal{I}, A\subsetneq\emptyset} \,;\, \forall A, n_A \geq 0 \;\&\; \forall i, \sum_{A: i\in A} n_A \leq C_i \right\}.$$

where n_A is the number of subsets A currently borrowed by the customers. Let e_A denote the vector of $\mathcal{N}$ whose coordinate A is equal to 1, and others are 0.

We introduce the projection ψ on space $\mathcal{X} = \{0, ..., C_1\} \times \{0, ..., C_2\} \times \cdots \times \{0, ..., C_I\}$:

$$\psi : \left\{ \begin{array}{ccc} \mathcal{N} & \longrightarrow & \mathcal{X} \\ n = (n_A)_{A\subset\mathcal{I}} & \longmapsto & x = \left(\sum_{A: i\in A} n_A\right)_{i\in\mathcal{I}} \end{array} \right.$$

The total number of items of each type in queues (i.e. currently used by the customers) is given by a vector $x = (x_1, \dots, x_I) \in \mathcal{X}$, where x_i is the number of items of type i. We consider the product order $\leq$ on $\mathcal{X}$.

We have two different types of transitions. For each $n \in \mathcal{N}$, and for each $A \subset \mathcal{I}$:

- There is a demand for subset A, with rate λ_A. The new state is: $n + e_{A^{(n)}}$, where
$$A^{(n)} = \{i \in A : (\psi(n))_i < C_i\} \tag{2}$$
denotes the items of set A that are available in state n and that are sent together to the customer.
- If $n_A > 0$, there is a joint service of A, with rate $\mu \cdot n_A$. The new state is $n - e_A$.

By a standard uniformization procedure, we can transform the above continuous time Markov chain to a discrete time Markov chain. The outgoing rate for each state is smaller than $\lambda + \mu\sum_{i\in\mathcal{I}} C_i$. To simplify notation and without loss of generality, we assume that the following condition holds:

$$\lambda + \mu \sum_{i\in\mathcal{I}} C_i = 1, \tag{3}$$

and we take the uniformization constant to be equal to 1.

We now explain briefly the discrete event representation of our (uniformized) Markov chain. Precisions for services are given in Appendix A.

1) Arrivals: For any $A \subset \mathcal{I}$, $A \neq \emptyset$, let d_A be the event of probability λ_A that corresponds to a "joint arrival to queues in A". We give the transition function g of the Markov chain X on $\mathcal{N}$ for an arrival d_A, $A \subset \mathcal{I}$, $A \neq \emptyset$, when the state is $n \in \mathcal{N}$:

$$g(n, d_A) = n + e_{A^{(n)}}, \tag{4}$$

where $A^{(n)} \subset A$ is defined by (2).

2) Services: Unfortunately, if services are not well chosen, the supremum chain does not move with any service. Indeed, let us consider a possible service a. For $g^{\sup}(x, a)$ to be different from x on the i-th coordinate, $1 \leq i \leq I$, we have that *all* states $n \in \mathcal{N}$ such that $\psi(n) = x$ must be served on some coordinate $A \subset \mathcal{I}$ that contains i (A can depend on n). Otherwise, if there is at least one state n whose i-th queue is not served, then the i-th coordinate of the supremum does not move. This makes the definition of services a little tricky; we

give the details in Appendix A.

B. Another stopping condition

Note that Algorithm 3 only gives an interval that contains the projection of a state distributed according to the stationary distribution. We can relax further the stopping condition: Instead of stopping when the upper and lower envelopes meet ($m = M$), we can stop when they meet at least once on each component between time $-t+1$ and time 0, see Algorithm 4. In fact, we will provide some bounds for the stopping time of Algorithm 4 (see Theorem 5), and compare Algorithms 3 and 4 with simulations (see IV-C).

Algorithm 4: Modified stopping condition

Data: *I.i.d.* events $(a_{-t})_{t\in\mathbb{N}} \in \mathcal{E}^{\mathbb{N}}$.
Result: Interval $[m^*, M^*] \subset \mathcal{X}$ containing the projection of a state $n^* \in \mathcal{N}$ distributed according to to the stationary distribution.

```
begin
    t := 1; c := zeros(1, I);
    repeat
        m := ⊥ (∈ X); M := ⊤ (∈ X);
        for i = t − 1 downto 0 do
            [m, M] := H([m, M], a_{−i}) ;
            for j = 1 to I do
                if m(j) = M(j) then c(j) := 1;
        t := 2t;
    until c = ones(1, I) ;
    m* := m; M* := M;
    return m*, M*;
end
```

We can give bounds for the complexity of Algorithm 4, assuming the following conditions. We suppose there exist two subsets $\mathcal{I}'_0$ and $\mathcal{I}'_C \subset \mathcal{I}$, $\mathcal{I} = \mathcal{I}'_0 \cup \mathcal{I}'_C$, such that:

(i) $\mu > \sum_{i\in\mathcal{I}'_0} \lambda_i$,

(ii) $\delta_p \stackrel{\text{def}}{=} \lambda_p - \mu\left(\sum_{i=1}^{p} C_i - 1\right) > 0$ for all $p \in \mathcal{I}'_C$.

Without loss of generality, we can change the numbering of queues such that:

(iii) $\left(i \in \mathcal{I}'_0 \text{ and } j \in \mathcal{I}'_C\right) \implies i \leq j$.

Theorem 5. *Assume conditions (i) to (iii) hold. Then we can bound the time τ_{Alg} for which all components couple at least once by:* $\mathbb{E}[\tau_{Alg}] \leq \frac{1}{\mu - \sum_{i\in\mathcal{I}'_0}\lambda_i} \sum_{i\in\mathcal{I}'_0} C_i + \sum_{p\in\mathcal{I}'_C} \frac{1}{\delta_p} C_p$.

The proof is given in Appendix E, and uses the results of Appendixes B, C, and D.

C. Numerical results

In Figure 1 on the top, we give stopping times for Algorithms 3 and 4 (ATO-POS with joint services), for the following parameters: $I = 5$, $C_i = 10, \forall i$, $\lambda_A = \frac{1}{2^{|A|-1}}$, and $\mu_i(x_i) = \mu x_i$, with $\rho = \frac{\lambda_i}{\mu}$. The size of the sample is $N = 100$. We can observe that the mean stopping times of both algorithms are very close. In Figure 1 on the bottom, we provide mean distance between upper and lower bounding states at time 0 using 1-norm, *i.e.* $\sum_{i\in\mathcal{I}}(M_i^* - m_i^*)$.

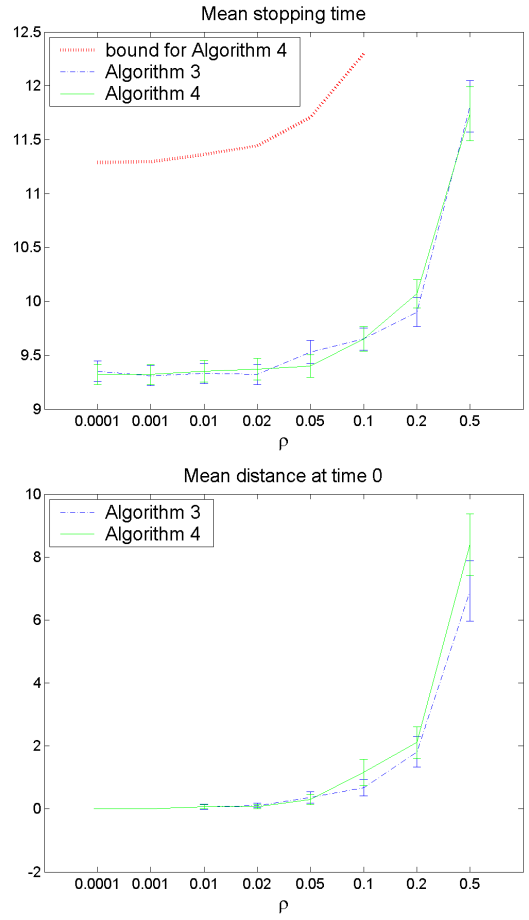

Fig. 1. On the top: Stopping times for Algorithms 3 and 4 (ATO-POS with joint services), and the upper bound for Algorithm 4 (we display $\log_2(T)$ where T is the mean stopping time), together with the 95% confidence intervals. On the bottom: Mean distance (in 1-norm) between upper and lower bounding states at time 0 for Algorithms 3 and 4, together with the 95% confidence intervals.

V. ATO-POS SYSTEM WITH JOINT SERVICES: HIGH SERVICE RATE CASE

The supremum chain $Y^{\sup}$ is monotone (Proposition 8, Appendix B). This gives directly the computation for the upper envelope of H: for any $m, M \in \mathcal{X}, a \in \mathcal{E}$, $\overline{H}([m, M], a) = g^{\sup}(M, a)$. This result and Lemma 4 give that the projected chain $\psi(X)$ is between zero and the supremum chain $Y^{\sup}$. If the service rate is high, we can wait until the supremum chain $Y^{\sup}$ hits zero. The main advantage is that this provides some solution for the possible decoupling of the system: When $Y^{\sup}$ reaches zero, the projected process is also in state zero, and thus the *only* possible state for the original Markov chain X is also zero. Thus the original chain X couples in $\mathcal{N}$. Hence, we can do backward simulation for $Y^{\sup}$ until we find a time $-t$ such that $Y^{\sup}_{-t} = 0$, and then, from time $-t$ to time 0, simulate the only trajectory of X starting from state zero (with the same events). We will refer to this algorithm as Algorithm 5.

Lemma 11 in Appendix B gives that the expected hitting time of zero for $Y^{\sup}$ is $O\left(\sum_i C_i\right)$, provided that $\mu > \sum_i \lambda_i$.

This result leads to the following theorem (since the *only* possible state for the chain X when $\psi(X) = 0$ is the state 0: hence if the projected process reaches zero, the original system defined on $\mathcal{N}$ also reaches zero, and so couples).

Theorem 6. *Let τ be the coupling time of the original chain X defined on $\mathcal{N}$, and corresponding to the ATO-POS system with joint services. Assume $\mu > \sum_i \lambda_i$. Then we have:*

$$\mathbb{E}[\tau] \leq \frac{1}{\mu - \sum_i \lambda_i} \sum_{i=1}^{I} C_i.$$

Furthermore, Lemma 11 shows that Algorithm 5 has linear complexity in $\sum_{i=1}^{I} C_i$, when $\mu > \sum_i \lambda_i$.

VI. Further remarks and conclusions

Up to our knowledge, this is the first time that the aggregation of Markov chains is combined with perfect sampling technique to avoid state space explosion problems. This direction sounds promising for various applications.

Note that, in the case of ATO-POS systems with joint services the state space is extremely big - only its dimension is $2^I - 1$ - and up to our best knowledge, there is no known efficient solution technique in the literature for this case. Thus our perfect sampling algorithms can be of great interest to evaluate their performance, as well as replace different existing approximation techniques used in the optimization algorithms for capacity dimensioning. Also, one possible natural future research direction in that area seems to be to perform an extensive study to evaluate accuracy of these approximations. Further, in most applications, the lost probability is demanded to be very low, thus the conditions in Section V seem to be natural, under which the exact samples of the stationary distribution can be obtained by Algorithm 5.

References

[1] J. G. Propp and D. B. Wilson, "Exact sampling with coupled Markov chains and applications to statistical mechanics," *Random Structures and Algorithms*, vol. 9, no. 1-2, pp. 223–252, 1996.

[2] O. Häggström and K. Nelander, "Exact sampling from anti-monotone systems," *Statist. Neerlandica*, vol. 52, no. 3, pp. 360–380, 1998.

[3] J.-M. Vincent, "Perfect simulation of monotone systems for rare event probability estimation," in *Winter Simulation Conference*, Orlando, 2005.

[4] W. S. Kendall and J. Møller, "Perfect simulation using dominating processes on ordered spaces, with application to locally stable point processes," *Advances in Applied Probability*, vol. 32, no. 3, pp. 844–865, 2000.

[5] A. Bušić, B. Gaujal, and J.-M. Vincent, "Perfect simulation and non-monotone markovian systems," in *Valuetools'08*, Athens, Grece, 2008.

[6] A. Bušić, B. Gaujal, and F. Pin, "Perfect sampling of Markov chains with piecewise homogeneous events," 2010, submitted. Preprint arXiv:1012.2910.

[7] F. P. Kelly, "Loss networks," *The Annals of Applied Probability*, vol. 1, no. 3, pp. 319–378, 1991.

[8] G. Louth, M. Mitzenmacher, and F. Kelly, "Computational complexity of loss networks," *Theoretical Computer Science*, vol. 125, no. 1, pp. 45–59, 1994.

[9] I. M. H. Vliegen and G. J. van Houtum, "Approximate evaluation of order fill rates for an inventory system of service tools," *International Journal of Production Economics*, vol. 118, no. 1, pp. 339–351, 2009.

[10] A. Bušić, I. Vliegen, and A. Scheller-Wolf, "Comparing Markov chains: Aggregation and precedence relations applied to sets of states, with applications to assemble-to-order systems," HAL, Tech. Rep., 2009.

[11] J.-S. Song and P. Zipkin, "Supply chain operations: Assemble-to-order systems," in *Supply Chain Management: Design, Coordination and Operation*, ser. Handbooks in Operations Research and Management Science, A. de Kok and S. Graves, Eds. North-Holland, 2003, vol. 11, ch. 11, pp. 561–596.

[12] J.-S. Song, S. H. Xu, and B. Liu, "Order-fulfillment performance measures in an assemble-to-order system with stochastic leadtimes," *Operations Research*, vol. 47, no. 1, pp. 131–149, 1999.

[13] S. Dayanik, J.-S. Song, and S. H. Xu, "The effectiveness of several performance bounds for capacitated production, partial-order-service, assemble-to-order systems," *Manufacturing & Service Operations Management*, vol. 5, no. 3, pp. 230–251, 2003.

[14] C. G. Cassandras and S. Lafortune, *Introduction to discrete event systems*, 2nd ed. Springer, 2008.

[15] D. A. Levin, Y. Peres, and E. L. Wilmer, *Markov chains and mixing times*. Providence, RI: American Mathematical Society, 2009, with a chapter by James G. Propp and David B. Wilson.

[16] P. Brémaud, *Markov chains: Gibbs fields, Monte Carlo simulation, and queues*. New York: Springer-Verlag, 1999.

Appendix

A. Events: definition of services

Before defining services, we need to define an ordering for the non-empty subsets of $\mathcal{I}$.

Ordering of the subsets. For all $i \in \mathcal{I}$, we define $\left(A_k^i\right)_{0 \leq k \leq 2^{I-i}-1}$ as an ordering of all the subsets of $\{i, ..., I\}$ containing i (for instance, subsets $\left(A_k^1\right)_k$ are those that contain 1, $\left(A_k^2\right)_k$ those that contain 2 but not 1, and so on). More precisely, let $i \in \mathcal{I}$ and $k \in \{0, 1, ..., 2^{I-i} - 1\}$. We set $k = k_1 \, ... \, k_{I-i}$ for the binary coding of k ($k = \sum_{s=1}^{I-i} k_s 2^{I-i-s}$, where $k_s \in \{0, 1\}$). Then the subset A_k^i is by definition such that: i) $i \in A_k^i$, and ii) for $s \in \{i+1, ..., I\}$, $s \in A_k^i$ if and only if $k_{s-i} = 0$.

Services. Let r_j^i, $1 \leq i \leq I$, $1 \leq j \leq C_i$, be independent events of probability μ to occur, such that the transition function g of X is given by:

- For each $n \in \mathcal{N}$, if $\sum_{\ell=0}^{2^{I-i}-1} n_{A_\ell^i} < j$, set: $g(n, r_j^i) = n$;
- For each $n \in \mathcal{N}$, if $\sum_{\ell=0}^{2^{I-i}-1} n_{A_\ell^i} \geq j$, let k be the minimal element of $\{0, 1, ..., 2^{I-i} - 1\}$ such that $\sum_{\ell=0}^{k-1} n_{A_\ell^i} < j \leq \sum_{\ell=0}^{k} n_{A_\ell^i}$. Set: $g(n, r_j^i) = n - e_{A_k^i}$.

We can notice that the total number of events corresponding to services is only: $\sum_{i=1}^{I} C_i$.

B. Monotonicity of the supremum chain

1) Computation of the supremum chain:

Lemma 7. *For all $x \in \mathcal{X}$, set $\hat{x}_i \stackrel{\text{def}}{=} \max\{x_i - (x_1 + ... + x_{i-1}), 0\}$. Then the transition function of the supremum chain is, for $A \subset \mathcal{I}$, $A \neq \emptyset$, $i \in \mathcal{I}$, $1 \leq j \leq C_i$, and $x \in \mathcal{X}$:*

$$\begin{cases} g^{\sup}(x, d_A) &= x + \sum_{k \in A} \mathbb{1}_{\{x_k < C_k\}} e_k, \\ g^{\sup}(x, r_j^i) &= x - \mathbb{1}_{j \leq \hat{x}_i} e_i. \end{cases}$$

2) Monotonicity: As a corollary of Lemma 7, we have the following result (the proof for services follows from the fact that $x \leq y$ and $x_i = y_i$ imply $\hat{x}_i \geq \hat{y}_i$).

Proposition 8. *Let $a \in \mathcal{E}$ be any arrival or service. With the natural order on $\mathcal{X}$, the event a is monotone for the supremum*

chain in the ATO-POS system with joint services, that is to say: for all $x, y \in \mathcal{X}$ such that $x \leq y$, we have that: $g^{\sup}(x,a) \leq g^{\sup}(y,a)$.

C. The aggregated envelope chain H

Let $m, M \in \mathcal{X}$, $m \leq M$. Our goal is to compute $H([m,M],a)$ for $a \in \mathcal{E}$ (see the definition in III-B).

Arrivals. Let $A \subset \mathcal{I}$, $A \neq \emptyset$. Then $H([m,M],d_A) = [f(m,d_A), f(M,d_A)]$ where $f(x,d_A) = g^{\sup}(x,d_A) = x + \sum_{k\in A} \mathbb{1}_{\{x_k < C_k\}} e_k = g^{\inf}(x,d_A)$ for all $x \in \mathcal{X}$. Indeed, the computation of $g^{\sup}(x,d_A)$ is given in Lemma 7, and the same proof can be applied to compute $g^{\inf}(x,d_A)$. In addition, the expression of $f(\cdot,d_A)$ shows that it is monotone, which gives the result for H.

Services. Let $i \in \mathcal{I}$ and $j \in \{1,2,...,C_i\}$. By Proposition 8, $g^{\sup}$ is monotone, thus $\overline{H}\left([m,M],r_j^i\right) = g^{\sup}(M,r_j^i) = M - \mathbb{1}_{j \leq \hat{M}_i} e_i$. In order to compute $\underline{H}\left([m,M],r_j^i\right)$, we first need to compute the infimum chain $g^{\inf}(\cdot,r_j^i)$ (Lemma 9). We will see that it is not monotone, hence we also need to compute the lower envelope of the infimum chain (Lemma 10).

1) Computation of the infimum chain for services:

Lemma 9. *We compute the p-th coordinate of $g^{\inf}(x,r_j^i)$, $p \in \mathcal{I}$, and we have to distinguish three cases:*

- *If $p < i$, then $\left(g^{\inf}(x,r_j^i)\right)_p = x_p$;*
- *If $p = i$, then $\left(g^{\inf}(x,r_j^i)\right)_p = x_i - \mathbb{1}\{j \leq x_i\}$;*
- *If $p > i$, then $\left(g^{\inf}(x,r_j^i)\right)_p = x_p - \mathbb{1}\left\{x_p > 0 \ \&\ j \leq \min\left(\sum_{i'=i+1}^p x_{i'}, x_i\right)\right\}$.*

2) Computation of the lower envelope for the infimum chain (for services): The computation above leads to the following observation: services are not monotone for the infimum chain. Indeed, let us take $I = 2$, $x = (0,1)$, and $y = (1,1)$. Then $x \leq y$, yet $g^{\inf}(x,r_1^1) = (0,1) \geq (0,0) = g^{\inf}(y,r_1^1)$. Thus, we have to compute the lower envelope of the infimum chain $Y^{\inf}$.

Lemma 10. *Let m, $M \in \mathcal{X}$ such that $m \leq M$. Let $i \in \mathcal{I}$ and $j \in \{1,2,...,C_i\}$. Set $m' = \underline{H}\left([m,M],r_j^i\right)$. We compute each coordinate p of m', for $p \in \mathcal{I}$, and we can distinguish three cases:*

- *If $p < i$, then $m'_p = m_p$;*
- *If $p = i$, then $m'_p = m_i - \mathbb{1}\{j \leq m_i\}$;*
- *If $p > i$, then $m'_p = m_p - \mathbb{1}\left\{m_p > 0 \ \&\ j \leq \min\left(\sum_{i'=i+1}^{p-1} M_{i'} + m_p, M_i\right)\right\}$.*

D. Hitting time to zero for the supremum chain

The next lemma gives the mean hitting time to zero for the supremum chain, using the results of Appendix B. We use it in the proofs of both Theorems 5 (IV-B) and 6 (V).

Lemma 11. *Let $\bar{\tau}$ be the time that the supremum chain, starting from $C = (C_1,...,C_I)$, reaches the state 0 in $\mathcal{X}$. Assume $\mu > \sum_i \lambda_i$. Then we have: $\mathbb{E}[\bar{\tau}] \leq \frac{1}{\mu - \sum_i \lambda_i} \sum_{i=1}^I C_i$.*

To prove this, we will use the following result that is often used as a part of the proof of Foster's theorem (see for instance [16, proof of Theorem 1.1]). It gives a bound on the mean hitting time of a subset for a Markov chain that verifies the following assumptions:

Theorem 12. *Let the transition matrix P on the finite state space E be irreducible and suppose that there exists a function $h : E \to \mathbb{R}_+$ such that*

$$\sum_{z\in E} P(y,z)h(z) \leq h(y) - \epsilon \text{ for all } y \notin U, \tag{5}$$

for some subset $U \subset E$. Let τ_U be the hitting time of U and $\mathbb{E}_y$ denote the expectation, knowing that the chain starts in y. Then, for all $y \notin U$,

$$\mathbb{E}_y[\tau_U] \leq \frac{h(y)}{\epsilon}. \tag{6}$$

Proof of Lemma 11. Let P be the transition matrix of the supremum chain whose transition function is given in Lemma 7.

We use Theorem 12, with $E = \mathcal{X}$, $U = \{0\}$ and $h(z) := \sum_{i=1}^I z_i$ for all $z \in \mathcal{X}$. For all $y \in \mathcal{X} \setminus \{0\}$, we have: $\sum_{z\in\mathcal{X}} P(y,z)h(z) = \sum_A \lambda_A \left(\sum_i y_i + \sum_{i\in A} \mathbb{1}\{y_i < C_i\}\right)$ $+\mu \sum_i \hat{y}_i \left(\sum_j y_j - 1\right) + \mu \sum_i (C_i - \hat{y}_i) \sum_j y_j$ $= \sum_i y_i + \sum_A \sum_{i\in A} \mathbb{1}\{y_i < C_i\}\lambda_A - \mu \sum_i \hat{y}_i$ $\leq h(y) + \sum_A |A|\lambda_A - \mu \sum_i \hat{y}_i = h(y) + \sum_i \lambda_i - \mu \sum_i \hat{y}_i$, where the second equality comes from equation (3). Hence we proved that, for all $y \in \mathcal{X}\setminus\{0\}$, $\sum_{z\in\mathcal{X}} P(y,z)h(z) \leq h(y) - \delta$, where $\delta = \mu \min_{y\neq 0} \sum_i \hat{y}_i - \sum_i \lambda_i$. In addition $\delta > 0$ since $\mu \cdot \min_{y\neq 0} \sum_i \hat{y}_i = \mu > \sum_i \lambda_i$ by hypothesis. Hence the condition (5) of Theorem 12 is proved: we can apply (6) with $y = (C_1,...,C_I)$. It follows that $\mathbb{E}[\bar{\tau}] = \mathbb{E}_y[\tau_{\{0\}}] \leq \frac{h(y)}{\delta} = \frac{1}{\delta}\sum_{i=1}^I C_i$, since the time for the system to hit zero is the time for state $y = (C_1,...,C_I)$ to hit zero, due to monotonicity (Proposition 8).

E. Proof of Theorem 5

The proof Theorem 5 is based on Lemma 11 and the following lemma, that gives a bound on the mean hitting time of C_p for the p-th coordinate of the infimum:

Lemma 13. *Let $p \in \mathcal{I}$, and assume $\delta_p := \lambda_p - \mu\left(\sum_{i=1}^p C_i - 1\right)$ is positive. Let $\tau^{(p)}$ be the time for the p-th coordinate of $\underline{H}$ to hit C_p (starting from 0). Then: $\mathbb{E}\left[\tau^{(p)}\right] \leq \frac{1}{\delta_p} C_p$.*

Proof of Theorem 5. Thanks to condition (iii), the projection of H on $\mathcal{I}'_0$ is a Markov chain. Condition (i) allows to apply Lemma 11 to this Markov chain. Condition (ii) allows to apply Lemma 13, and the fact that $\mathcal{I} = \mathcal{I}'_0 \cup \mathcal{I}'_C$ concludes the proof. ∎