



HAL
open science

Perfect sampling of stochastic matching models with reneging

Thomas Masanet, Pascal Moyal

► **To cite this version:**

Thomas Masanet, Pascal Moyal. Perfect sampling of stochastic matching models with reneging. 2022.
hal-03580942v1

HAL Id: hal-03580942

<https://hal.science/hal-03580942v1>

Preprint submitted on 25 Feb 2022 (v1), last revised 18 Jan 2024 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

PERFECT SAMPLING OF STOCHASTIC MATCHING MODELS WITH RENEGING

THOMAS MASANET

IECL, Université de Lorraine / INRIA PASTA

PASCAL MOYAL

IECL, Université de Lorraine / INRIA PASTA

ABSTRACT. In this paper, we introduce a slight variation of the Dominated Coupling From the Past algorithm (DCFTP) of Kendall, for bounded Markov chains. It is based on the control of a (typically non-monotonic) stochastic recursion by a (typically monotonic) one. We show that this algorithm is particularly suitable for stochastic matching models with bounded patience, a class of models for which the steady state distribution of the system is in general unknown in closed form. We first show that the Markov chain of this model can be easily controlled by an infinite-server queue. We then investigate the particular case where patience times are deterministic, and this control argument may fail. In that case we resort to an ad-hoc technique that can also be seen as a control (this time, by the arrival sequence). We then compare this algorithm to the classical CFTP one, and show how our perfect simulation results can be used to estimate, and compare, the loss probabilities of various systems in equilibrium.

1. INTRODUCTION

The study of stochastic matching models is currently a very active line of research in applied probability. It has been demonstrated in various contexts, that these stochastic models are suitable to capture the dynamics of a wide range of real-time random systems, in which elements enter the system at (possibly) random times, with a view to finding a *match*, that is identified as such following specified compatibility rules, given by a given compatibility graph between classes of items. Then, matched couples leave the system right away as soon as they found a match. This is the case in various applications, such as, peer-to-peer applications, job-search, public housing or college allocations, organ transplants, blood banks, car sharing, assemble-to-order systems, and so on. These models have been introduced in [14] for bipartite graphs (which is suitable for supply/demands-type applications) and arrivals by couples, as a variant of the seminal works [15, 3]. To account for a wider range of applications (e.g., dating websites, crossed kidney transplants, assemble-to-order systems or car-sharing), they have been generalized to general graphs (with simple arrivals) in [26], and then to hypergraphs in [31, 33] and graphs with self-loops, in [7].

Applications such as organ transplants are subject to very strong timing constraints: the patients waiting for a transplant have a finite life time in the system, and similarly, available organs are highly perishable, and must be transplanted very quickly. Hence the need to incorporate an impatience (or renegeing) parameter to

the system. More precisely, in this paper we address a general stochastic matching model, as defined in [26], in which the elements have a finite (and possibly random) patience upon arrival, before the end of which they must find a match. Otherwise, they renege and leave the system forever. Matching models with impatience have recently been addressed for a bipartite model and the ‘N’ graph in [16] for a matching policy of the ‘First Come, First Matched’ (FCFM) type, and from the point of view of stochastic optimization, for partially static policies, in [5]. On another hand, in [22], stability conditions, together with moment bounds at equilibrium, have been given for models in which some, but not all, classes of items are impatient, and the matching policy is of the ‘Max-weight’ class.

It is important, however, to observe that the stationary distributions of stochastic matching models are in general unknown, and that little is known about the characteristics of the steady state. Models implementing the FCFM policy constitute an exception, in which the stationary state can often be characterized in a product form, using dynamic reversibility arguments (see, along the various models, [1, 2, 7, 30, 17]), for models without reneging. However, in the cases of models with reneging, aside from the particular graph geometries addressed in [16], no exact results are known. Moreover, FCFM policies are clearly not always the best option in a real-time context: coming back to the case of organ transplants, other criteria must be taken into account, such as the level of emergency, equity, ages of the patients/donor, various levels of compatibilities, and so on. Mimicking the various existing results in queueing theory, implementing policies of the ‘Match the Longest queue’ (ML) or ‘Earliest Deadline First’ (EDF) type may be profitable to minimize loss, and it is significant that EDF does *not* amount to FCFM if the patience times are random.

Our aim is to analyse matching models with reneging in steady state, for general matching policies. In view of the above discussion, we thus need to assess the stationary distribution of the matching model at hand, without knowledge of this distribution in closed form. As is well known, this task can be handled by *simulating perfectly*, this steady state.

Perfect simulation has been a constantly active line of research in the analysis of stochastic systems, since the pioneering works of Propp and Wilson [32], and Borovkov and Foss [10, 11]. The underlying idea is now well known: Consider a discrete-event stochastic system whose stationary distribution is intractable mathematically. Then we can study the system in steady state, by *precisely* simulating samples of the stationary distribution, even though the latter is not known in closed form, instead of approximating it by long-run trajectories. Then, various average performance parameters at equilibrium can be assessed by Monte-Carlo techniques.

The celebrated Propp and Wilson algorithm [32] is based on coupling-from-the-past (CFTP), namely, all trajectories of the considered Markov chain coalesce before time 0, whenever these trajectories are initiated from all possible states of the chain, far away enough in the past. This phenomenon is closely related to the concept of strong backwards coupling (see e.g. [12] or Chapter 2.5 of [6]), and the connections between the two notions are investigated for various cases of stochastic recursions in [19]. Strong backwards coupling is the pillar of the construction of the stationary state under general non-Markov assumptions, *via* the use of renovating events, see e.g. [10, 11]. It is also a tool to construct stationary states on enriched probability spaces, via skew-product constructions, see [25, 4, 29].

As they rely on the exact coalescence of a family of Markov chains, CFTP algorithms are typically adapted to finite-state spaces and to monotonic dynamics, using envelope techniques. Various authors have extended these settings: generalizing the ideas in [19], it is proven in [24] that geometrically ergodic Markov

chains admit a CFTP algorithm of the envelope-type, even if they are not monotonic, a result that was then generalized to a wider class of ergodic Markov chains, in [18]. Various related approaches have then been proposed, that all rely on the following intuitive idea: simulating from the past a more ‘simple’ recursion, and deducing the stationary state of the recursion of interest by comparison. This is the core idea of the so-called ‘Dominated coupling from the past’ (DCFTP) introduced in [24], of the so-called ‘Bounding chains’ of Huber [20, 21], that are particularly adapted to mechanical-statistical contexts, or of various envelope techniques for queueing systems, see e.g. [13]. More recently, DCFTP-related methods have been implemented, together with saturation techniques, to perfectly simulate non-Markov queueing systems, see [8] for infinite-server and loss queues, and [9] for multiple-server queues.

This paper is a first contribution to the perfect sampling of stochastic matching models. We introduce two perfect sampling algorithms, Algorithms 2 and 3 below, that produce samples of the stationary distribution of stochastic matching models with reneging, in the case where arrivals are discrete. The first algorithm simply relies on the control of the model at hand by an infinite server queue, an algorithm that would clearly not be optimal in a context of heavy traffic. Indeed, as was observed in [8], as it relies on the depletion of a corresponding infinite-server model, the coalescence time for Algorithm 2 grows exponentially in function of the arrival rates, see [23]. Our second algorithm, Algorithm 3, is peculiar to the case where patience times are deterministic (and so the matching policies FCFM and EDF are equivalent). In that case, we propose an ad-hoc control of the system simulated backwards in time by the input of the system. Then, the performances in terms of complexity and coalescence time, are way better than the usual CFTP. In fact, both Algorithms 2 and 3 can be seen as particular cases of a more general perfect sampling algorithm for bounded Markov chains, Algorithm 1 below, which we call perfect sampling *by control*, a condition that is closely related to those under which a DCFTP-type algorithm can be implemented.

This paper is organized as follows. After some preliminary in Section 2, we introduce our general perfect sampling algorithm by control in Section 3. In Section 4, we introduce the general stochastic matching model with reneging, and the two corresponding perfect sampling algorithms in sub-sections 4.2 and 4.3.2. The performances of the latter algorithm are investigated in sub-section 4.3.3, and a first application to the comparison of the steady-state performances of two matching policy (here, EDF (or in other words FCFM) and ML), is provided in sub-section 4.3.5.

2. PRELIMINARY

In what follows, \mathbb{R} , \mathbb{N} and \mathbb{Z} denote respectively the sets of real, non-negative integers and relative integers, respectively. We also let \mathbb{X} and \mathbb{V} denote two sets. All considered random variables (r.v.’s, for short) are defined on a common probability space $(\Omega, \mathcal{F}, \mathbb{P})$.

Definition 1. *Let \mathbb{X} and \mathbb{V} be two sets. Let $k \in \mathbb{Z}$ and $x \in \mathbb{X}$. Let f be a mapping from $\mathbb{X} \times \mathbb{V}$ to \mathbb{X} , and $(v_n)_{n \in \mathbb{Z}}$ be an identically distributed sequence of \mathbb{V} -valued r.v.’s. We denote by $(X_n^k(x))_{n \geq k}$, the stochastic recursive sequence (SRS) driven by $(f, (v_n)_{n \in \mathbb{Z}})$, of initial value x at time k . Namely, $(X_n^k(x))_{n \geq k}$ is fully determined by the recurrence equation*

$$\begin{cases} X_k^k(x) & = x; \\ X_{n+1}^k(x) & = f(X_n^k(x), v_n), \quad \text{a.s. for all } n \geq k. \end{cases}$$

It is immediate that $(X_n^k(x))_{n \in \mathbb{Z}}$ is a \mathbb{X} -valued Markov chain whenever the sequence $(v_n)_{n \in \mathbb{Z}}$ is IID. Conversely, any \mathbb{X} -valued Markov chain $(Z_n)_{n \in \mathbb{N}}$ of fixed starting time k and initial value x , having transition matrix Q over \mathbb{X} , can be represented by the SRS driven by $(f, (v_n)_{n \in \mathbb{Z}})$, where $(v_n)_{n \in \mathbb{Z}}$ is an IID sequence of uniformly distributed r.v.'s on $[0, 1]$ and f is piecewise constant and satisfies for all $x_1, x_2 \in \mathbb{X}$,

$$\lambda(\{x : f(x_1, x) = x_2\}) = Q(x_1, x_2),$$

for λ is the Lebesgue measure, see e.g. Section 2.5.3 of [6].

Fix an SRS $X := (X_n^k(x))_{n \geq k}$. Then for all $e \in \mathbb{X}$, we set

$$\tau_e^{X,k}(x) = \inf \{n \geq k : X_n^k(x) = e\},$$

the hitting time of value e by $(X_n^k(x))_{n \geq k}$. If $(v_n)_{n \in \mathbb{Z}}$ is IID, then $(X_n^k(x))_{n \geq k}$ is a Markov chain, and the distribution of $\tau_e^{X,k}(x) - k$ is independent of k . In that case, we then denote by $\tau_e^X(x)$, a generic r.v. that is so distributed.

3. A PERFECT SAMPLING ALGORITHM BY CONTROL

In this section we present a perfect simulation algorithm, Algorithm 1, for processes that are bounded, in a sense that we will make precise hereafter, closely related to the dominated coupling from the past algorithm introduced by Kendall in [18]. Until the end of this section, we fix three sets \mathbb{X}, \mathbb{Y} and \mathbb{V} , and two mappings $f : \mathbb{X} \times \mathbb{V} \rightarrow \mathbb{X}$ and $g : \mathbb{Y} \times \mathbb{V} \rightarrow \mathbb{Y}$.

Algorithm 1 Simulation of the stationary probability of X

Require: $a_1, \dots, a_q \in \mathbb{X}$, $b_1, \dots, b_q, y \in \mathbb{Y}$, a probability distribution ν on \mathbb{V}

```

 $r \leftarrow -1$ 
 $r_0 \leftarrow 0$ 
 $Y \leftarrow y$ 
while  $Y \notin \{b_1, \dots, b_q\}$  do
   $i \leftarrow r$ 
   $Y \leftarrow y$ 
  for  $j = r_0, \dots, r$  do
     $v_j \sim \nu$ 
  end for
  while  $i > 0$  and  $Y \notin \{b_1, \dots, b_q\}$  do
     $Y \leftarrow g(Y, v_i)$ 
     $i \leftarrow i + 1$ 
  end while
   $r_0 \leftarrow r + 1$ 
   $r \leftarrow 2r$ 
end while
for  $k = 1, \dots, q$  do
  if  $Y = b_k$  then
     $X \leftarrow a_k$ 
  end if
end for
while  $i > 0$  do
   $X \leftarrow f(X, v_i)$ 
   $i \leftarrow i + 1$ 
end while
return  $X$ 

```

Let us first observe the following,

Lemma 1. *Let $y, b_1, \dots, b_q \in \mathbb{Y}$, $(v_n)_{n \in \mathbb{Z}}$ be a stationary \mathbb{V} -valued sequence distributed from ν , and let Y be a \mathbb{Y} -valued SRS driven by $(g, (v_n)_{n \in \mathbb{Z}})$. Suppose that*

$$(1) \quad \max_{i \in \llbracket 1, q \rrbracket} \mathbb{P} [\tau_{b_i}^Y(y) < \infty] = 1.$$

Then, Algorithm 1 terminates almost surely.

Proof. Let $i \in \llbracket 1, p \rrbracket$ such that $\mathbb{P} [\tau_{b_i}^Y(y) < \infty] = 1$. We show that

$$\mathbb{P} \left(\bigcup_{n \in \mathbb{N}} \left\{ \tau_{b_i}^{Y, -2^n}(y) \leq 0 \right\} \right) = 1.$$

But for any $N \in \mathbb{N}$, we get that

$$\begin{aligned} \mathbb{P} \left(\bigcup_{n \in \mathbb{N}} \left\{ \tau_{b_i}^{Y, -2^n}(y) \leq 0 \right\} \right) &= \mathbb{P} \left(\bigcup_{n \in \mathbb{N}} \left\{ \tau_{b_i}^{Y, -2^n}(y) + 2^n \leq 2^n \right\} \right) \\ &\geq \mathbb{P} \left(\left\{ \tau_{b_i}^{Y, -2^N}(y) + 2^N \leq 2^N \right\} \right) = \mathbb{P} (\{ \tau_{b_i}^Y(y) \leq 2^N \}), \end{aligned}$$

in view of the stationarity of the input. Thus we obtain that

$$\begin{aligned} \mathbb{P} \left(\bigcup_{n \in \mathbb{N}} \left\{ \tau_{b_i}^{Y, -2^n}(y) \leq 0 \right\} \right) &\geq \lim_{N \rightarrow +\infty} \mathbb{P} (\{ \tau_{b_i}^Y(y) \leq 2^N \}) \\ &= \mathbb{P} (\{ \tau_{b_i}^Y(y) < +\infty \}) = 1, \end{aligned}$$

as desired. \square

3.1. A control condition.

Definition 2. *Let $(X_n)_{n \in \mathbb{Z}}$ and $(Y_n)_{n \in \mathbb{Z}}$ be two SRS, respectively valued in \mathbb{X} and \mathbb{Y} and $q \in \mathbb{N}^*$. We say that $(Y_n)_{n \in \mathbb{Z}}$ q -controls $(X_n)_{n \in \mathbb{Z}}$, if there exists $b_1, \dots, b_q, y \in \mathbb{Y}$ and $a_1, \dots, a_q \in \mathbb{X}$ such that*

$$(2) \quad \forall i \in \llbracket 1, q \rrbracket, \forall k \in \mathbb{Z}, \forall n \geq k, [Y_n^k(y) = b_i] \implies [\forall x \in \mathbb{X}, X_n^k(x) = a_i].$$

b_1, \dots, b_q are called the endpoints of Y . If $q = 1$ we simply say that Y controls X .

The following result establishes that under certain conditions including the control of $(X_n)_{n \in \mathbb{Z}}$ and $(Y_n)_{n \in \mathbb{Z}}$, Algorithm 1 terminates almost surely, and the output is a sample of the stationary distribution of the SRS $(X_n)_{n \in \mathbb{Z}}$.

Theorem 1. *Suppose that the sequence $(v_n)_{n \in \mathbb{Z}}$ is IID, and let X and Y be two SRS respectively driven by $(f, (v_n)_{n \in \mathbb{Z}})$ and $(g, (v_n)_{n \in \mathbb{Z}})$. Suppose that Y is positive recurrent, and that X is q -controlled by Y . Then the Markov chain X is itself positive recurrent, and its output is sampled from the stationary distribution of X .*

Proof. Suppose that Y q -controls X , and denote b_1, \dots, b_q, y and a_1, \dots, a_q the corresponding parameters. As the chain Y is positive recurrent we get that $\mathbb{E} (\tau_{b_i}^Y(y)) < \infty$ for all $i \in \llbracket 1, q \rrbracket$. But by assumption we get that

$$\forall x \in \mathbb{X}, \forall i \in \llbracket 1, q \rrbracket, \quad \mathbb{E} (\tau_{a_i}^X(x)) \leq \mathbb{E} (\tau_{b_i}^Y(y)),$$

and so X is itself positive recurrent.

By assumption, (1) holds. So from Lemma 1, Algorithm 1 terminates almost surely. We now show that its output is precisely that of the CFTP algorithm for the chain X , of starting times $N = -1, -2, -4, \dots$. Let N be the backwards

coalescence time of the chain X , that is, the smallest starting time for which the CFTP algorithm terminates for X , or in other words

$$(3) \quad N = \min \{n \geq 0 : X_0^{-n}(x) = X_0^{-n}(x') \text{ for all } x \neq x' \in \mathbb{X}\}.$$

Let R be the smallest termination time of Algorithm 1. Then, by the very definition of Algorithm 1 and (2) there exists $i \in \llbracket 1, q \rrbracket$, a time $n_0 > 0$ such that $-R < -n_0$, and such that $X_{-n_0}^{-R}(x) = X_{-n_0}^{-R}(x') = a_i$ for all $x, x' \in \mathbb{X}$, $x \neq x'$, and thereby

$$(4) \quad X_0^{-R}(x) = X_0^{-R}(x') = X_0^{-n_0}(a_i), \text{ for all } x, x' \in \mathbb{X}, x \neq x'.$$

In particular, it naturally follows from (3) that we necessarily have $R \geq N$, otherwise all versions of the chain X starting from a time posterior to $-N$, would have coalesced before 0, an absurdity. Therefore, by the definition of CFTP we get that $X_0^{-R}(x) = X_0^{-N}(x)$ for all $x \in \mathbb{X}$. In other words, Algorithm 1 and the CFTP algorithm produce the same output which, from (4), is given by $X_0^{-n_0}(a_i)$. But it is well known that the CFTP algorithm produces a sample of the stationary distribution of X over \mathbb{X} . This completes the proof. \square

Assumption (2) is key to our analysis. Under this control condition the value of the SRS Y forces that of X at time n , whatever the initial value of X at time N . This is reminiscent of the concept of renovating event, as introduced by Borovkov and Foss, see [10, 11]. Let us remind the following,

Definition 3. Let X be an SRS driven by f and $(v_n)_{n \in \mathbb{Z}}$, and let A be a subset of \mathbb{X} . We say that $(A_n)_{n \in \mathbb{N}}$ is a sequence of renovating events of length m and associated mapping $h : \mathbb{V}^m \rightarrow \mathbb{X}$ for the chain X if for any $n \in \mathbb{Z}$, on A_n we have

$$X_{n+m} = h(v_n, \dots, v_{n+m-1}).$$

Now suppose that (2) holds for $a_1, \dots, a_q, b_1, \dots, b_q$ and y . Then, it is easily seen that for all $k \in \mathbb{Z}$, $i \in \llbracket 1, q \rrbracket$ and $x \in \mathbb{X}$, $(\{Y_n^k(y) = b_i\})_{n \geq k}$ is a sequence of renovating events of length 1 for the sequence $(X_n^k(x))_{n \geq k}$. Indeed, for all $n \geq k$, on $\{Y_n^k(y) = b_i\}$ we get that $X_n^k(x) = a_i$, and therefore

$$X_{n+1}^k(x) = f(a_i, v_n) =: h(v_n).$$

3.2. The ordered case. A typical context in which the control of the SRS X by the SRS Y occurs, is when the two sequences are constructed on the same input, and their driving maps satisfy some monotonicity properties, which we detail below. Throughout this section, (E, \prec, o) denotes a partially ordered space, of minimal point o (i.e., such that $o \prec e$ for all $e \in E$), and we define two mappings $\varphi : \mathbb{X} \mapsto E$ and $\psi : \mathbb{Y} \mapsto E$ such that

$$|\varphi^{-1}(o) \cap \mathbb{X}| = |\psi^{-1}(o) \cap \mathbb{Y}| = 1.$$

We also suppose that there exists $y \in \mathbb{Y}$, such that

$$(5) \quad \forall x \in \mathbb{X}, \varphi(x) \prec \varphi(y).$$

Definition 4. We say that the mapping $f : \mathbb{X} \rightarrow E$ is dominated (for φ and ψ) by the mapping $g : \mathbb{Y} \rightarrow E$, and denote $f \prec^{\varphi, \psi} g$, if

$$\forall x \in \mathbb{X}, y \in \mathbb{Y}, [\varphi(x) \prec \psi(y)] \implies [\varphi \circ f(x) \prec \psi \circ g(y)].$$

Proposition 1. In the case $\mathbb{X} = \mathbb{Y}$, we have $f \prec^{\varphi, \psi} g$ under either one of conditions (1) and (2) below,

- (1) $\psi \circ g$ is \prec -nondecreasing and uniformly lower-bounded by $\varphi \circ f$;
- (2) $\varphi \circ f$ is \prec -nondecreasing and uniformly upper-bounded by $\psi \circ g$.

Proof. Plainly, if we assume that (1) holds, for all $x, y \in \mathbb{X}$ such that $\varphi(x) \prec \psi(y)$ we get that $\varphi \circ f(x) \prec \psi \circ g(x) \prec \varphi \circ g(y)$, whereas if (2) holds we obtain that $\varphi \circ f(x) \prec \varphi \circ f(y) \prec \psi \circ g(y)$. \square

Proposition 2. *Let X and Y be two SRS, respectively driven by $(f, (v_n)_{n \in \mathbb{Z}})$ and $(g, (v_n)_{n \in \mathbb{Z}})$, where the input $(v_n)_{n \in \mathbb{Z}}$ is IID on \mathbb{V} . Suppose that $f(\cdot, v) \prec^{\varphi, \psi} g(\cdot, v)$ for all $v \in \mathbb{V}$, and that there exists $y \in \mathbb{Y}$ such that $\varphi(x) \prec \psi(y)$ for all $x \in \mathbb{X}$. Then, if the SRS Y is positive recurrent, so is the SRS X . Moreover, Algorithm 1 for y , $a = \varphi^{-1}(o) \cap \mathbb{X}$ and $b = \psi^{-1}(o) \cap \mathbb{Y}$ terminates a.s., and produces a sample of the stationary distribution of X .*

Proof. We aim at showing that X controls Y for b, y, a . Let $k, n \in \mathbb{Z}$ be such that $n > k$ and $Y_n^k(y) = b$. Let $x \in \mathbb{X}$. We show by induction on ℓ , that for all $\ell \in \llbracket k, n \rrbracket$,

$$(6) \quad \varphi(X_\ell^k(x)) \prec \psi(Y_\ell^k(y)).$$

First, from (5) we get that $\varphi(X_k^k(x)) = \varphi(x) \prec \psi(y) = \psi(Y_k^k(y))$, so (6) holds for $\ell = k$. Suppose that it is true at rank $\ell \in \llbracket k, n-1 \rrbracket$. Let $c = Y_\ell^k(y)$ and $d = X_\ell^k(x)$, in a way that $\varphi(d) \prec \psi(c)$, by the induction assumption. Then, from the domination assumption of f by g we obtain that

$$\varphi(X_{\ell+1}^k(x)) = \varphi(f(d, v_\ell)) \prec \psi(g(c, v_\ell)) = \psi(Y_{\ell+1}^k(y)),$$

so (6) holds at rank $\ell + 1$. It is therefore true for all $\ell \in \llbracket k, n \rrbracket$. In particular, we have that $\varphi(X_n^k(x)) \prec \psi(Y_n^k(y)) = \psi(b) = o$, implying that $X_n^k(x) = a$. Thus Y controls x , and we conclude using Theorem 1. \square

As a conclusion, provided that $f \prec^{\varphi, \psi} g$, Algorithm 1 provides a perfect sampling algorithm for the SRS X . In fact, in this ordered case Algorithm 1 is closely related to the DCFTP algorithms of Kendall, see [24, 18]. Specifically, as in [18] we have an upper bound process Y that we will simulate backward in time. But as this process is bounded, it does not have to be at equilibrium. We also have a lower bound process, namely the constant process equal to $\psi^{-1}(o) \cap \mathbb{Y}$. Similarly to the sandwiching method in [32], we only have to simulate the process Y starting at state y . When that process meets the lower bound backward in time, means that coalescence has been detected. Then, as in [24, 18], we simulate X starting from a single state until time 0.

A similar idea is used for the perfect sampling of loss queueing systems in [8] using the domination of the system by an infinite server queue in some sense (an idea that we also in the construction of Section 4.2 below) and likewise, for the perfect sampling of multiple-server queues in [9].

Remark 1. The above DCFTP conditions are in fact reminiscent of stochastic domination conditions for the construction of stationary SRS's in the general stationary ergodic context. For instance, for $\mathbb{X} = E$ a lattice space and for $\varphi = \psi = Id$, Condition (1) in Proposition 1 above amounts to condition (H1) in [29] for any SRS X and Y that are respectively driven by f and g , and a common input $(v_n)_{n \in \mathbb{Z}}$. This latter condition guarantees, under general stationary ergodic assumptions, the existence of a stationary version of the SRS X , at least on an extended probability space, provided that a stationary version of the SRS Y exists on the original one, see Theorem 3 in [29] (see also [25]).

4. A STOCHASTIC MATCHING MODEL WITH RENEGING

In this section we address the perfect sampling of the stationary state of a class of models, which we refer to as ‘general stochastic matching models with impatience’.

4.1. The model. We consider a general stochastic matching model (GM), as was defined in [26]: items enter one by one in a system, and each of them belongs to a determinate class. The set of classes is denoted by V , and identified with $\{1, \dots, |V|\}$. We fix a simple, connected graph $G = (V, E)$ having set of nodes V , termed *compatibility graph*. Upon arrival, any incoming item of class, say, $i \in V$ is either matched with an item present in the buffer, of a class j such that i shares an edge with j in G , if any, or if no such item is available, it is stored in the buffer to wait for its match. Whenever several possible matches are possible for an incoming item i , a *matching policy* determines what is the match of i without ambiguity. Each matched pair departs the system right away.

A GM model with impatience is a GM model in which each entering item in the system is assigned a patience time upon arrival. If the considered item has not been matched at the end of her patience time, then she leaves the system forever. To formalize this, after fixing the compatibility graph $G = (V, E)$, and the matching policy Φ , we set $\xi = 1$ as the interarrival time of items (i.e., there is an arrival at each unit of time) and two IID sequences $(V_n)_{n \in \mathbb{Z}}$ and $(P_n)_{n \in \mathbb{Z}}$, where for all n and $P_n \in \mathbb{R}_+$ and $V_n \in V$ respectively represent the class and the patience time of the n -th item entering the system. The two sequences $(V_n)_{n \in \mathbb{Z}}$ and $(P_n)_{n \in \mathbb{Z}}$ are not necessarily independent. In particular, it can be the case that the patience time P_n of the n -th item depends on her class V_n .

The class of models defined in Section 4.1 admits the following Markov representation. Define the set

$$\mathbb{X} := \{\emptyset\} \cup \bigcup_{q=1}^{\infty} (\mathbb{R}_+^* \times V)^q.$$

For all $t \geq 0$, let $Q(t)$ be the number of customers in the system at time t , and let us define the *profile* of the system at t , as the following element of \mathbb{X} ,

$$(7) \quad X(t) = \begin{cases} ((R^1(t), V^1(t)), \dots, (R^{Q(t)}(t), V^{Q(t)}(t))) & \text{if } Q(t) \geq 1, \\ \emptyset & \text{else,} \end{cases}$$

where for all $i \in \llbracket 1, Q(t) \rrbracket$, we denote by $R^i(t)$ (resp., $V^i(t)$) the remaining patience at time t (resp., the class) of the i -th item in line at time t , in the order of arrivals. If the system is empty at t , we again set $X(t) = \emptyset$.

Definition 5. *We say that the matching policy Φ is admissible if, upon each arrival, the choice of the match amongst compatible items in line at t , if any, is made according to the sole knowledge of $X(t)$, and possibly of a draw that is independent of everything else.*

Remark 2. It is easily seen that matching policies that depend only on the arrival times (*First Come, First Matched*, denoted hereafter by FCFM, or *Last Come, First Matched*), remaining patience times (*Earliest Deadline First, Latest Deadline First*), matching policies that depend on the queue sizes of the various nodes (*Match the Longest, Match the Shortest, Max-Weight*) and priority policies (see e.g. [26, 22] for details) are all admissible.

Set $(T_n)_{n \in \mathbb{Z}} = (n)_{n \in \mathbb{Z}}$, the arrival times to the system, and for all $n \in \mathbb{Z}$, denote by $X_n = X(T_n^-) = X(n^-)$, the state of the system seen by the customer entered at time n . Then we obtain the following result,

Proposition 3. *For any admissible matching policy Φ , the profile sequence $(X_n)_{n \in \mathbb{Z}}$ is stochastic recursive, driven by the triplet sequence $((V_n, P_n))_{n \in \mathbb{Z}}$, and a mapping $f^\Phi : \mathbb{X} \times (\mathbb{R}_+ \times V) \mapsto \mathbb{X}$ that depends on Φ and possibly on a random draw*

independent of everything else. In other words we get that

$$X_{n+1} = f^\Phi(X_n, (P_n, V_n)), \quad n \in \mathbb{Z}.$$

Proof. The construction of f^Φ is immediate: If the incoming element at n is matched upon arrival, the couple corresponding to its match, determined by Φ , is erased from the vector X_n ; else, the couple (V_n, P_n) is added at the end of the vector X_n . Last, the couples (possibly including the incoming couple (V_n, P_n)) whose second coordinate is strictly less than 1 at n are erased from the vector X_n (because they will have reneged by time $n + 1$), and the second coordinates of all other couples of X_n , if any, decrease by 1. \square

4.2. A first Perfect sampling algorithm. We can then design a first perfect sampling algorithm for matching models with impatience, that is simply based on the control (in the sense of Section 3) by an infinite server system. Specifically, denote by V and P , two random variables having respective distributions, the generic distributions of $(V_n)_{n \in \mathbb{Z}}$ and $(P_n)_{n \in \mathbb{Z}}$. Let $X := (X_n)_{n \in \mathbb{Z}}$ be the patience profile SRS defined above. In this context, we let $Y := (Y_n)_{n \in \mathbb{Z}}$ be a \mathbb{R}_+ -valued SRS defined by the recursion

$$(8) \quad Y_{n+1} = [\max(Y_n, P_n) - 1]^+ =: g(Y_n, (V_n, P_n)), \quad n \in \mathbb{Z}.$$

Combining Lemma 5 of [27] with Corollary 2 in [28], we get that whenever

$$(9) \quad \mathbb{P}(P < \xi) > 0,$$

the Markov chain $(Y_n)_{n \in \mathbb{Z}}$ is positive recurrent. Consider the following algorithm, Algorithm 2, which is a declination of Algorithm 1 started with $y = m$, for Y the recursion defined by (8), $q = 1$, $a_1 = \emptyset$ and $b_1 = 0$. We have the following result,

Theorem 2. *Under condition (9), the profile Markov chain $(X_n)_{n \in \mathbb{Z}}$ is positive recurrent. If moreover there exists $m > 0$ such that $\mathbb{P}(P \leq m) = 1$, then Algorithm 2 terminates almost surely, and its output is sampled from the stationary distribution of $(X_n)_{n \in \mathbb{Z}}$.*

Proof. We apply Proposition 2, by setting in this case

$$(10) \quad \begin{array}{lll} \varphi : & \mathbb{X} & \longrightarrow \mathbb{R}_+ \\ & x = ((r_1, v_1), \dots, (r_q, v_q)) \neq \emptyset & \longmapsto \max \{r_i : i \in \llbracket 1, q \rrbracket\} \\ & \emptyset & \longmapsto 0. \end{array}$$

As any item spends in the system a time that is less or equal to its patience time, for any $n \in \mathbb{Z}$, $\varphi(X_n)$ corresponds to the largest remaining maximal sojourn time in the system of an item in the system just before time T_n . Consequently, for any $(p, v) \in \mathbb{R}_+ \times V$, for all $x \in \mathbb{X}$ we obtain that

$$(11) \quad \varphi(f^\Phi(x, (p, v))) \leq [\max(\varphi(x), p) - 1]^+ = g(\varphi(x), p).$$

Therefore, for any $x \in \mathbb{X}$ and $y \in \mathbb{R}_+$ such that $\varphi(x) \leq y$, Then, as $g(\cdot, (z, p))$ is non-decreasing on \mathbb{R}_+ we get that

$$\varphi(f^\Phi(x, (p, v))) \leq g(y, p).$$

Proposition 2 completes the proof. \square

4.3. Deterministic patience times. Whenever the condition (9) does not hold, the process keeping track of the largest remaining sojourn times in the system fails to dominate the primary Markov chain $(X_n)_{n \in \mathbb{N}}$ of the system. In fact, the very positive recurrence of the system is not granted. One then has to resort to ad-hoc techniques to show stability, and to sample the stationary state.

In this section, we consider the particular case of constant patience time $P := p \in \mathbb{N}^*$, so that (9) fails. For short, we denote such a matching model by (G, Φ, μ, p) . We

Algorithm 2 Simulation of the stationary probability of X - Matching model with impatience

Require: a probability distribution ν on $V \times \mathbb{R}_+$

```

 $r \leftarrow -1$ 
 $r_0 \leftarrow 0$ 
 $Y \leftarrow m$ 
while  $Y \neq 0$  do
   $i \leftarrow r$ 
   $Y \leftarrow y$ 
  for  $j = r_0, \dots, r$  do
     $(v_j, p_j) \sim \nu$ 
  end for
  while  $i > 0$  and  $Y \neq 0$  do
     $Y \leftarrow [\max(Y, p_i) - 1]^+$ 
     $i \leftarrow i + 1$ 
  end while
   $r_0 \leftarrow r + 1$ 
   $r \leftarrow 2r$ 
end while
if  $Y = 0$  then
   $X \leftarrow \emptyset$ 
end if
while  $i > 0$  do
   $X \leftarrow f^\Phi(X, (v_i, p_i))$ 
   $i \leftarrow i + 1$ 
end while
return  $X$ 

```

show that such systems are always positive recurrent, and construct an alternative perfect sampling algorithm that is another declination of Algorithm 1, and is again based on the control condition defined in Section 3. In this particular case, the profile Markov chain can be simplified, so as to obtain the following alternate, simpler, Markov representation of the system state,

Definition 6. For all $n \in \mathbb{Z}$, the word-profile of the system just before time n is defined by the word

$$\tilde{X}_n = w_1 \cdots w_p \in (V \cup \{0\})^p,$$

where for all $i \in \llbracket 1, p \rrbracket$,

$$w_i = \begin{cases} V_{n-p+i-1} & \text{if the item entered at } n-p+i-1 \text{ was not matched before } n; \\ 0 & \text{else.} \end{cases}$$

In particular, if the element entered at time $n-p$ is still in the system at time n (its class thus appearing as the first letter of the word \tilde{X}_n), it is either matched with the incoming element at time n , or immediately lost. We call $\tilde{\mathbb{X}} \subset (V \cup \{0\})^p$, the (finite) state space of \tilde{X} . Similarly to Proposition 3, it is immediate that for any admissible policy Φ , the sequence $(X_n)_{n \in \mathbb{Z}}$ is Markov, and we denote by \tilde{f}^Φ , the (deterministic, up to a possible draw that is independent of everything else) map $\tilde{f}^\Phi : \tilde{\mathbb{X}} \times V \rightarrow \tilde{\mathbb{X}}$, such that

$$\tilde{X}_{n+1} = \tilde{f}^\Phi(\tilde{X}_n, V_n), \quad n \in \mathbb{Z}.$$

4.3.1. *Synchronizing words.* For a fixed model (G, Φ, μ, p) , for any word $v = v_1 \cdots v_p$ in V^* and any $\tilde{X} \in \tilde{\mathbb{X}}$, let us denote by $W^\Phi(\tilde{X}, v) \in \tilde{\mathbb{X}}$, the state of a system started at \tilde{X} and receiving the arrivals v_1, \dots, v_p in that order.

Definition 7. Fix a model (G, Φ, μ, p) . A word $w = w_1 \cdots w_q \in V^*$ is said to be synchronizing, if

$$\exists z \in \tilde{\mathbb{X}}, \forall \tilde{X} \in \tilde{\mathbb{X}}, W^\Phi(\tilde{X}, w) = z.$$

In other words, v is a synchronizing word if all buffers synchronize to some value \tilde{X} when they are fed by a common arrival scenario v , whatever the initial state. It is obvious how synchronizing words can be used for perfect simulation. Indeed, if we start Markov chains at a time $-M$ from all possible states, observing a synchronizing word of length $q < M$ amongst the arrivals (in the sense that the classes of q consecutive incoming items are given by the letters of w , in that order), clearly guarantees that all chains have coalesced. We first prove the existence of synchronizing words for any discrete matching system, by providing a sufficient condition for this.

Definition 8. Let $w \in V^*$. We say that the word of length $2p$, $w = w_1 \cdots w_{2p} \in V^*$ satisfies the p -condition if

$$\forall i \in \llbracket 1, p \rrbracket, \forall j \in \llbracket p+1, p+i \rrbracket, w_i \not\sim w_j.$$

Theorem 3. In a discrete matching model with impatience (G, Φ, μ, p) , any word that satisfies the p -condition is a synchronizing word.

Proof. Let $w = w_1 \cdots w_{2p}$ be a word satisfying the p -condition, and let $u = w_1 \cdots w_p$ and $v = w_{p+1} \cdots w_{2p}$. Let $\tilde{X} \in \tilde{\mathbb{X}}$ and $\mathbf{0}_p = \underbrace{0 \cdots 0}_p$, be the empty state. As u is of

length p , any item present in the buffer represented by \tilde{X} is no longer in there after the arrivals represented by u . Therefore $W^\Phi(\tilde{X}, u) = u' = w'_1, \dots, w'_p$ where for all $i \in \llbracket 1, p \rrbracket$, $w'_i = w_i$ if the corresponding item is still in the buffer after these arrivals, or $w'_i = 0$ else. As w satisfies the p -condition, for any $i \in \llbracket 1, p \rrbracket$ such that $w'_i \neq 0$ and any $j \in \llbracket p+1, p+i \rrbracket$, we have that $w'_i \not\sim w_j$. All elements corresponding to non-zero letters of u' are not matched, because their patience necessarily expires before the arrival of a compatible item, and no letters from v can be married to a letter in u' . Therefore if $j, h \in \llbracket p+1, 2p \rrbracket$ are such that the element corresponding to w_j is matched to that corresponding to w_h if we add v to the empty buffer $\mathbf{0}_p$, then it is also the case if we add v to the buffer u' . In other words, we get that

$$W^\Phi(\tilde{X}, w) = W^\Phi(W^\Phi(\tilde{X}, u), v) = W^\Phi(u', v) = W^\Phi(\mathbf{0}_p, v).$$

As this is true for any $\tilde{X} \in \tilde{\mathbb{X}}$, w is a synchronizing word for $z := W^\Phi(\mathbf{0}_p, v)$. \square

We proceed with two technical lemmas. In what follows, for all $a \in V \cup \{0\}$ and all $k \in \llbracket 0, p \rrbracket$ we define the following word of length p ,

$$x^a(k) = \underbrace{0 \cdots 0}_k \underbrace{a \cdots a}_{p-k}.$$

First observe the following,

Lemma 2. Consider a matching model with impatience (G, FCFM, μ, p) , with matching policy FCFM. Let $a \in V$. Then, for all $k \in \llbracket 0, p-1 \rrbracket$, for all words w of length p , $W^\Phi(x^a(k), w)$ and $W^\Phi(x^a(k+1), w)$ differ at most by one letter in some position i (substituting 0 to the i -th letter).

Proof. Let $k \in \llbracket 0, p-1 \rrbracket$, and write $w = w_1 \cdots w_p$. With some abuse, in the proof below the matching procedure of the initial state $x^a(k)$ (or $x^a(k+1)$) with the arrival represented by w is itself called $W^{\text{FCFM}}(x_k^a, w)$ (or $W^{\text{FCFM}}(x^a(k+1), w)$).

If $w_i \neq a$ for all $i \in \llbracket 1, p \rrbracket$, then we trivially get that

$$W^{\text{FCFM}}(x^a(k), w) = W^{\text{FCFM}}(x^a(k+1), w).$$

Else, let i_1, \dots, i_l be the indices, in increasing order, of the letters of w matched with letters of $x^a(k+1)$ in $W^{\text{FCFM}}(x^a(k+1), w)$. There are three possibilities for the indices (in increasing order) of the letters of w that are matched with letters of x_k^a in $W^{\text{FCFM}}(x_k^a, w)$ (which we call for short “the indices” in the discussion hereafter):

- (1) Either the first a of x_k^a is matched in $W^{\text{FCFM}}(x_k^a, w)$ with a letter of w of index $i_0 < i_1$. Then all the remaining a 's in x_k^a are matched in $W^{\text{FCFM}}(x_k^a, w)$ exactly as the a 's in $W^{\text{FCFM}}(x^a(k+1), w)$, and so the indices are precisely i_0, i_1, \dots, i_l .
- (2) Or the first a of x_k^a is not matched in $W^{\text{FCFM}}(x_k^a, w)$. Then, all the remaining a 's of $x^a(k)$ are matched in $W^{\text{FCFM}}(x_k^a, w)$ exactly as the a 's in $W^{\text{FCFM}}(x^a(k+1), w)$, and the indices are again i_1, \dots, i_l .
- (3) Or, the first matched a of x_k^a in $W^{\text{FCFM}}(x_k^a, w)$ is matched with the letter of index i_1 in w . Then, in $W^{\text{FCFM}}(x_k^a, w)$, either the indices of the matched letters of w are the same as in $W^{\text{FCFM}}(x^a(k+1), w)$ (and then the last a in $x^a(k)$ remains unmatched), or the first $p-k-1$ a 's of x_k^a are matched with letters of w at indices i_1, \dots, i_l , and the last a is matched with a letter of w of index i_{l+1} , with $i_l < i_{l+1}$, in which case the indices are i_1, \dots, i_{l+1} .

If the indices are i_1, \dots, i_l , then $W^{\text{FCFM}}(x^a(k), w) = W^{\text{FCFM}}(x^a(k+1), w)$. If the indices are i_0, i_1, \dots, i_l or i_1, \dots, i_{l+1} then there is a letter b of w that is not matched with an a of $x^a(k+1)$ in $W^{\text{FCFM}}(x^a(k+1), w)$, but is matched with an a of x_k^a in $W^{\text{FCFM}}(x_k^a, w)$. Then, either that letter b remains unmatched in $W^{\text{FCFM}}(x^a(k+1), w)$, in which case $W^{\text{FCFM}}(x^a(k+1), w)$ and $W^{\text{FCFM}}(x^a(k), w)$ differ only at index i_0 (or i_{l+1}), where there is a b in $W^{\text{FCFM}}(x^a(k+1), w)$ and 0 in $W^{\text{FCFM}}(x^a(k), w)$. Or, b is matched with a letter c of w in $W^{\text{FCFM}}(x^a(k+1), w)$. Then, either the letter c remains unmatched in $W^{\text{FCFM}}(x^a(k), w)$, in which case $W^{\text{FCFM}}(x^a(k+1), w)$ and $W^{\text{FCFM}}(x^a(k), w)$ differ only at the place of that letter c in $W^{\text{FCFM}}(x^a(k), w)$, where there is a 0 in $W^{\text{FCFM}}(x^a(k+1), w)$. Or, c is matched with another letter b' in $W^{\text{FCFM}}(x^a(k), w)$, in which case we can repeat the same procedure for b' instead of b . As we have a finite number of letters in w , we eventually stop with a letter being present in a buffer and 0 in the other. In all cases, the buffers $W^{\text{FCFM}}(x^a(k+1), w)$ and $W^{\text{FCFM}}(x^a(k), w)$ differ only by one letter. \square

For all $\tilde{X} \in \tilde{\mathcal{X}}$, let us denote

$$T(\tilde{X}, a) = \text{Card} \left\{ \text{letters } \tilde{X}_i \text{ of } \tilde{X} : x_i - a \right\}.$$

It follows from the above that

Corollary 1. *Let w be a word of length $2p$ such that for some $i \in \llbracket 1; p \rrbracket$ and $j \in \llbracket p+1; p+i \rrbracket$, we have $w_i = w_j$. For such couple $\{i, j\}$, and $k \in \llbracket 0, 2p \rrbracket$, let*

$$x^{i,j}(k) = \begin{cases} x^{w_i}(k) & \text{if } k \in \llbracket 0, p-1 \rrbracket, \\ \mathbf{0}_{2p} & \text{if } k = p, \\ x^{w_j}(2p-k) & \text{if } k \in \llbracket p+1, 2p \rrbracket. \end{cases}$$

Let also $u(k) = W^{\text{FCFM}}(x^{i,j}(k), w_1 \cdots w_p)$, for all $k \in \llbracket 0, 2p \rrbracket$. Then, there exists an integer k in $\llbracket 0, 2p-1 \rrbracket$, such that $u(k) = z_1 \cdots z_p$ differs from $u(k+1) = z'_1 \cdots z'_p$ by only one letter in some position l , such that $z_l = w_j$, $z'_l = 0$, and for all $h \in \llbracket 1, p \rrbracket$, $z'_h = w_h$ or $z'_h = 0$. Moreover we have that $T(u(k), w_j) = 1$ and $T(u(k+1), w_j) = 0$.

Proof. By Lemma 2, for all we have that $k \in \llbracket 0, 2p - 1 \rrbracket$, $u(k)$ and $u(k + 1)$ differ at most by one letter in some position i (one being w_i , the other being a 0). Therefore, for all $k \in \llbracket 0, 2p - 1 \rrbracket$, $|T(u_k, w_{j^*}) - T(u_{k+1}, w_{j^*})| \leq 1$. Now notice that $2 \leq T(u(0), w_j)$, because gathering the words $x^{i,j}(0)$ and w would lead to at least $p + 1$ w_i 's out of $2p$ letters - so at least two w_i must remain in $u(0)$. On the other hand, we also have that $T(u(2p), w_j) = 0$, because any letter of $w_1 \cdots w_p$ that can be matched with w_j get matched in $u(2p)$ with the letters of $x^{i,j}(2p)$. As a consequence, there exists a rank $k \in \llbracket 0, 2p - 1 \rrbracket$ such that $T(u_k, w_j^*) = 1$ and $T(u_{k+1}, w_j^*) = 0$. The remaining statements follow readily from Lemma 2. \square

Theorem 4. *Consider a matching model with impatience (G, FCFM, μ, p) . Let w be a word of length $2p$ of V^* . The following conditions are equivalent:*

- (i) w satisfies the p -condition;
- (ii) w is a synchronizing word.

Proof. In view of Theorem 3, only the implication (i) \Rightarrow (ii) remains to be proven. For this, we reason by contraposition. So let w be a word of length $2p$ such that $w_i - w_j$ for some $i \in \llbracket 1; p \rrbracket$ and $j \in \llbracket p + 1; p + i \rrbracket$. Let $i^* \in \llbracket 1; p \rrbracket$ and $j^* \in \llbracket p + 1; p + i^* \rrbracket$ such that $w_{i^*} - w_{j^*}$ and $j^* = \inf\{j \in \llbracket p + 1; 2p \rrbracket, \exists i \in \llbracket 2p - j; p \rrbracket w_i - w_j\}$. We denote $u = w_1 \cdots w_p$ and $v = w_{p+1} \cdots w_{2p}$. Let k^* be the integer obtained in Corollary 1 for $i \equiv i^*$ and $j \equiv j^*$. Then, we get $u(k^*) = d_1 \cdots d_p$ and $u(k^* + 1) = e_1 \cdots e_p$, where $u(\cdot)$ is defined in Corollary 1. We will prove that $W^{\text{FCFM}}(u(k), v) \neq W^{\text{FCFM}}(u(k + 1), v)$, which will show in turn that w is not a synchronizing word.

Let i_1, \dots, i_l be the indices (in increasing order) of letters of v that are matched with letters of $u(k)$ in $W^{\text{FCFM}}(u(k), v)$, and i'_1, \dots, i'_h be the indices (in increasing order) of letters of v that are matched with letters of $u(k + 1)$ in $W^{\text{FCFM}}(u(k + 1), v)$. Now let us define the following sets,

$$\begin{cases} I_0 & = \emptyset; \\ I_{m+1} & = I_m \cup \{\inf\{j \in \llbracket p + 1, p + m + 1 \rrbracket \setminus I_m : w_j - d_{m+1}\}\}, m \in \llbracket 0, p - 1 \rrbracket. \end{cases}$$

At each step of this construction we add to the set I_m the index of the letter that is matched with d_{m+1} in $W^{\text{FCFM}}(u(k), v)$, if any, as in FCFM, d_{m+1} is matched with the first compatible letter that has not been matched to a previous letter of u_k . In particular, we finally obtain that $I_p = \{i_1, \dots, i_l\}$. In the same way, we define the sets

$$\begin{cases} I'_0 & = \emptyset; \\ I'_{m+1} & = I'_m \cup \{\inf\{j \in \llbracket p + 1, p + m + 1 \rrbracket \setminus I'_m : w_j - e_{m+1}\}\}, m \in \llbracket 0, p - 1 \rrbracket, \end{cases}$$

and the same argument leads to $I'_p = \{i'_1, \dots, i'_h\}$.

If in Corollary 1, the letter a that can be matched with w_j^* in $u(k)$ (and be replaced by a 0 in $u(k + 1)$) is at position m , then by construction of $j^*, u(k)$ and $u(k + 1)$, a will be matched with w_{j^*} in $u(k)$. So the m^{th} step is different for $W^{\text{FCFM}}(u(k), v)$ and $W^{\text{FCFM}}(u(k + 1), v)$. For every other step m' , as $d_{m'} \neq w_{j^*}$ and $e_{m'} \neq w_{j^*}$, we add the same letter, if any, to $I_{m'-1}$ and $I'_{m'-1}$. So we have $I_p = I'_p \cup \{j^*\}$. Let n_1 (resp., n_2) be the total number of letters from v that are matched in $W^{\text{FCFM}}(u(k), v)$ (resp., $W^{\text{FCFM}}(u(k + 1), v)$). As the total numbers of matched letters are even, both $n_1 + |I_p|$ and $n_2 + |I'_p|$ are even. But as $|I_p|$ and $|I'_p|$ are of different parity, so are n_1 and n_2 . Thus,

$$\begin{aligned} W^{\text{FCFM}}(x^{i^*j^*}(k^*), w) &= W^{\text{FCFM}}(u(k^*), v) \neq W^{\text{FCFM}}(u(k + 1), v) \\ &= W^{\text{FCFM}}(x^{i^*j^*}(k^* + 1), w), \end{aligned}$$

and w is not a synchronizing word. \square

We have proven that the p -condition is a necessary and sufficient condition for being a synchronizing word of length $2p$ in the case where the matching policy is FCFM. It is not the case for all matching policies. For example, for the matching policy LCFM (Last Come, First Matched'), it can be proven that there exists synchronizing words of length $\lfloor \frac{3p}{2} \rfloor$, so that any suffix of those words that would not satisfy the p -condition would still be a synchronizing word. However, as we prove hereafter, the p -condition is a simple criteria, that can be used to construct an efficient perfect sampling algorithm.

4.3.2. *A second perfect sampling algorithm.* We are now in position to introduce a perfect sampling algorithm for the state of a matching model with deterministic patience.

Definition 9. Consider a model (G, Φ, μ, p) , and define for all $k \in \mathbb{Z}$, the SRS $\tilde{Y} := (\tilde{Y}_n^k)_{n \geq k}$ on the set $\tilde{Y} = \{\emptyset\} \cup \bigcup_{j=1}^{2p} V^j$, as follows:

$$\begin{cases} \tilde{Y}_k^k &= \emptyset; \\ \tilde{Y}_{n+1}^k &= \tilde{g}(Y_{n+1}^k, v_{n+1}) \\ &:= \begin{cases} v_1 \cdots v_i V_{n+1}, & \text{if } Y_n^k = v_1 \cdots v_i \in V^i \text{ with } i < 2p \\ v_2 \cdots v_{2p} V_{n+1}, & \text{if } Y_n^k = v_1 \cdots v_{2p} \in V^{2p} \end{cases}, \quad n \geq k, \end{cases}$$

in a way that for all $n \geq k + 2p$, \tilde{Y}_n represents the last $2p$ arrivals to the system at time n .

Consider the following algorithm, Algorithm 3. It consists of another declination of Algorithm 1, started with $\tilde{Y} = \emptyset$, for \tilde{Y} the recursion of Definition 9, b_1, \dots, b_q , the words satisfying the p -condition, and a_1, \dots, a_q , the states of \tilde{X} after the arrival of b_1, \dots, b_q , respectively.

We have the following result,

Proposition 4. \tilde{X} is recurrent positive. Moreover, Algorithm 3 terminates a.s., and its output is sampled from the stationary distribution of \tilde{X} .

Proof. We can easily show that \tilde{Y} q -controls \tilde{X} , with q the number of word satisfying the p -condition. Let w be a word satisfying the p -condition. By Theorem 3, w is a synchronizing word. Thus for all $k \in \mathbb{Z}$ and $n \geq k$, we get in particular that

$$(12) \quad \left[\tilde{Y}_n^k(\emptyset) = w \right] \implies \left[\forall \tilde{x} \in \tilde{X}, \tilde{X}_n^k(\tilde{x}) = W^\Phi(\emptyset, w) \right],$$

which implies that \tilde{Y} controls \tilde{X} over all words satisfying the p -condition. We conclude using Theorem 1. \square

Remark 3. Observe that \tilde{Y} is not irreducible, however it reaches its recurrent class in $2p$ iterations. So for all w satisfying the p condition, we still have that

$$\mathbb{P}(\tau_\emptyset^{\tilde{Y}}(w) < \infty) = 1.$$

4.3.3. *Efficiency of Algorithm 1.* In this section we analyse the coalescence time of Algorithm 1. We first provide a bound for the average number of iterations of the algorithm to see the coalescence time, and then for the corresponding horizon in the past. We have the following,

Proposition 5. Let I be the number of iterations of Algorithm 1 to detect coalescence, and $T = -p^{2I}$ be the corresponding starting time. Then, we have that

$$\mathbb{E}[-T] \leq \frac{2p}{q^{p,\mu}},$$

Algorithm 3 Simulation of the stationary probability of \tilde{X} - Matching model with deterministic patience

Require: a probability distribution ν on V

```

 $r \leftarrow -1$ 
 $r_0 \leftarrow 0$ 
 $\tilde{Y} \leftarrow \emptyset$ 
while  $\tilde{Y} \notin \{b_1, \dots, b_q\}$  do
     $i \leftarrow r$ 
     $Y \leftarrow y$ 
    for  $j = r_0, \dots, r$  do
         $v_j \sim \nu$ 
    end for
    while  $i > 0$  and  $\tilde{Y} \notin \{b_1, \dots, b_q\}$  do
         $\tilde{Y} \leftarrow \tilde{g}(\tilde{Y}, v_i)$ 
         $i \leftarrow i + 1$ 
    end while
     $r_0 \leftarrow r + 1$ 
     $r \leftarrow 2r$ 
end while
for  $k = 1, \dots, q$  do
    if  $\tilde{Y} = b_k$  then
         $\tilde{X} \leftarrow a_k$ 
    end if
end for
while  $i > 0$  do
     $\tilde{X} \leftarrow \tilde{f}^\Phi(\tilde{X}, v_i)$ 
     $i \leftarrow i + 1$ 
end while
return  $X$ 

```

where

$$q^{p,\mu} = \mathbb{P}[V_1 \cdots V_{2p} \text{ satisfies the } p \text{ condition}].$$

Proof. For any integer $n \geq 1$, we let for all $i \in \mathbb{N}^*$, z_i^n be the word of length $2p$ representing the arrivals into the system between time $-p2^n + (i-1)2p$ and time $-p2^n + i2p - 1$ included, in the order of arrivals. We also let

$$K^n = \inf \{i \in \mathbb{N}^* : z_i^n \text{ satisfies the } p\text{-condition}\}.$$

The independence of arrivals implies that the r.v.'s K^n , $n \in \mathbb{N}^*$ are identically distributed (but not independent) of geometric distribution of parameter

$$q^{p,\mu} = \mathbb{P}[V_1 \cdots V_{2p} \text{ satisfies the } p \text{ condition}].$$

Now, it readily follows from Theorem 3, that for all $n \in \mathbb{N}^*$, $I \leq n$ in particular if there has been an arrival array satisfying the p -condition between times $-p2^n$ and -1 included, that is, if $2pK^n \leq p2^n$. Consequently, for all $n \in \mathbb{N}^*$ we get that

$$\mathbb{P}[-T > p2^n] = \mathbb{P}[I > n] \leq \mathbb{P}[2pK^n > p2^n] = \mathbb{P}[2pK^1 > p2^n].$$

This readily implies that $-T \leq_{\text{st}} 2pK^1$, where \leq_{st} denotes the strong stochastic ordering. We deduce that

$$\mathbb{E}[-T] \leq 2p\mathbb{E}[K^1] = \frac{2p}{q^{p,\mu}}.$$

□

Whenever the arrival measure μ is uniform over V , the latter results specializes as follows,

Corollary 2. *If the graph $G = (V, E)$ is of size n and μ is uniform over V , we get the bounds*

$$\mathbb{E}[-T] \leq \frac{2pn^{2p}}{N(G, p)}, \quad \mathbb{E}[I] \leq 1 + \frac{2p \text{Log} n - \text{Log} N(G, p)}{\text{Log} 2},$$

where $N(G, p)$ is the number of words of V^* satisfying the p -condition.

Proof. The results readily follow from Proposition 5, observing that in this case

$$q^{p, \mu} = \frac{N(G, p)}{n^{2p}}.$$

□

For a given G and a given p , computing the number $N(G, p)$ of words satisfying the p -condition, is of crucial interest to assess the efficiency of Algorithm 1. As Corollary 2 demonstrates, a function of the latter quantity provides bounds for the expected values of $|T|$ and I . We now turn to a specific evaluation of $N(G, p)$, and for this, we first need the following definition,

Definition 10. *Let $(G = (V, E), \Phi, \mu, p)$ be a discrete matching model with impatience. For any word $w = w_1 \cdots w_{2p}$ satisfying the p -condition, the vision of w is defined as the word Z^w gathering, in their order of apparences, all distinct letters of the second half of w . In other words, we set*

- (1) $Z_1^w = w_{p+1}$,
- (2) For all $i \in \llbracket 1, p-1 \rrbracket$,

$$Z_{i+1}^w = \begin{cases} Z_i^w, & \text{if } w_{p+i+1} \in Z_i^w; \\ Z_i^w w_{p+i+1}, & \text{if } w_{p+i+1} \notin Z_i^w, \end{cases}$$

and $Z^w \equiv Z_p^w$.

In what follows, for any word $z = z_1 \cdots z_l$, we denote by $\beta(z)$ the cardinality of the set of nodes that are incompatible with all letters of z , namely

$$\beta(z) = \text{Card} \left\{ v \in V : \forall i \in \llbracket 1, l \rrbracket, v \not\sim z_i \right\}.$$

We have the following,

Proposition 6. *Let $(G = (V, E), \Phi, \mu, p)$ be a discrete matching model with impatience, and let $\mathcal{H}(G)$ be the set of words having distinct letters, that form a permutation of a set $U \subset V$ that is such that $\mathcal{E}(U) \neq V$. Then, the number $N(G, p)$ of words satisfying the p -condition is given by*

$$N(G, p) = \sum_{z = z_1 \cdots z_l \in \mathcal{H}(G)} \sum_{\{1 = k_0 < k_1 < \cdots < k_{l-1} < k_l = p\}} \prod_{j=1}^l j^{k_j - k_{j-1} - 1} (\beta(z_1 z_2 \cdots z_j))^{k_j - k_{j-1}}.$$

Proof. For any word $z = z_1 \cdots z_l \in V^*$ having l distinct letters, and any word $w = w_1 \cdots w_{2p}$ of vision z , denote the integers

$$\begin{cases} k_0^w &= 1; \\ k_i^w &= \inf \left\{ j \in \llbracket p+1, 2p \rrbracket, w_j = z_{i+1} \right\}, \quad i \in \llbracket 1, l-1 \rrbracket; \\ k_l^w &= p. \end{cases}$$

Let $1 = k_0 < k_1 < \cdots < k_{l-1} < k_l = p$ be a fixed family of integers. For any $i \in \llbracket 1, l-1 \rrbracket$, there are $\prod_{j=1}^i j^{k_j - k_{j-1} - 1}$ possible permutations of the i first distinct letters z_1, \dots, z_i appearing in the second half of w such that $(k_0^w, k_1^w, \dots, k_i^w) =$

$(k_1, k_2, \dots, k_{i+1})$, as the letters of indices $p+k_0^w, p+k_1^w, \dots, p+k_i^w$ are fixed and there are j allowed letters between the indices $k_j^w - 1$ and k_j^w . Moreover, for any such i there are also $\prod_{j=1}^i \beta(z_1 z_2 \dots z_j)^{k_j^w - k_{j-1}^w}$ words of length k_i , having only letters that are incompatible with z_1, \dots, z_i . Therefore, for any fixed family of integers $1 = k_0 < k_1 < \dots < k_{l-1} < k_l = p$, the number of such words w satisfying $k_i^w = k_i$ for all i , equals $\prod_{j=1}^l (j\beta(z_1 z_2 \dots z_j))^{k_j - k_{j-1}}$, and so the number of words satisfying the p -condition and having z as a vision is given by

$$(13) \quad N_z := \sum_{\{1=k_0 < k_1 < \dots < k_{l-1} < k_l = p\}} \prod_{j=1}^l j^{k_j - k_{j-1} - 1} (\beta(z_1 z_2 \dots z_j))^{k_j - k_{j-1}}.$$

Last, to get $N(p)$ we must sum the above quantity over all possible visions of words satisfying the p -condition. To characterize this set, observe that any vision z necessarily has distinct letters, forming a permutation of set $U_z \subset V$. If $\mathcal{E}(U_z) \neq V$, then there exists a letter $i \in V \setminus \mathcal{E}(U_z)$, and z clearly is a vision for any word w whose prefix of size p is $ii \dots i$ and whose suffix of size p is a permutation of the elements of U . If now $\mathcal{E}(U_z) = V$, as for any word w satisfying the p -condition and having vision z we must have $w_p \notin \mathcal{E}(U_z)$, we get to an immediate contradiction. Thus it is necessarily and sufficient that $\mathcal{E}(U_z) \neq V$ for z to be a vision, which concludes the proof. \square

4.3.4. Example. To illustrate the efficiency of Algorithm 1 in the case of deterministic patience times, we consider the simple non-trivial example of the so-called *paw* graph G of Figure 1.

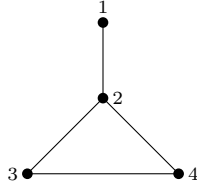


FIGURE 1. The paw graph.

As the above results demonstrate, for any p , to compute the number $N(G, p)$ of words satisfying the p -condition we first need to determine the set of all possible visions $\mathcal{H}(G)$ of G . In the present case we readily obtain that

$$\mathcal{H}(G) = \{1, 2, 3, 4, 13, 14, 31, 34, 41, 43, 134, 143, 314, 341, 413, 431\}.$$

It is then immediate to compute $\beta(z)$ for all $z \in \mathcal{H}(G)$ using (13). We obtain

$$\begin{cases} N_1 = 3^p, N_2 = 1, N_3 = 2^p, N_4 = 2^p, N_{13} = \frac{3}{2}4^p - 2 \cdot 3^p, N_{14} = \frac{3}{2}4^p - 2 \cdot 3^p, \\ N_{31} = \frac{1}{2}4^p - 2^p, N_{34} = (p-1)2^{p-1}, N_{41} = \frac{1}{2}4^p - 2^p, N_{43} = (p-1)2^{p-1}, \\ N_{134} = 3 \cdot 2^{2p-1} - (2p+4)3^{p-1}, N_{143} = 3 \cdot 2^{2p-1} - (2p+4)3^{p-1}, \\ N_{314} = 2^{2p-1} - 4 \cdot 3^{p-1} + 2^p, N_{341} = 2 \cdot 3^{p-1} - (p+1)2^{p-1}, \\ N_{413} = 2^{2p-1} - 4 \cdot 3^{p-1} + 2^p, N_{431} = 2 \cdot 3^{p-1} - (p+1)2^{p-1}. \end{cases}$$

Summing the above and rearranging, we obtain that

$$N(G, p) = 1 + 2^{2p+3} - 3^{p+1} - 4(p+3)3^{p-1}.$$

p	$N(G, p)$	Bound for $\mathbb{E}[I]$	Bound for $\mathbb{E}[-T]$
1	8	2	4
2	42	3,608	24,381
3	216	5,245	113,778
4	1050	6,964	499,322
5	4872	8,750	2152,250
6	21834	10,586	9220,784
7	95352	12,460	39412,874
8	408378	14,360	168274,189
9	1723176	16,283	717831,830
10	7187946	18,223	3059320,779

TABLE 1. Efficiency of Algorithm 1.

Then, applying Corollary 2 and Jensen's inequality we obtain the following bound for the average number of iterations of Algorithm 1 to detect coalescence,

$$\begin{aligned} \mathbb{E}[I] &\leq 1 + \frac{2p \operatorname{Log} n - \operatorname{Log} N(G, p)}{\operatorname{Log} 2} \\ &= 1 + 4p - \frac{\operatorname{Log}(1 + 2^{2p+3} - 3^{p+1} - 4(p+3)3^{p-1})}{\operatorname{Log} 2} =: B_I, \end{aligned}$$

and the average starting time T to detect coalescence is bounded by $-p2^{B_I}$. In Table 1, we specify the number of words satisfying the p condition, together with the corresponding bounds for $\mathbb{E}[I]$ and $\mathbb{E}[-T]$, for various values of the deterministic patience p .

Complexity comparison. After having provided a bound for the average coalescence time for algorithm 1, we now compare the number of operations necessary to complete Algorithm 1, to the number of operations necessary to complete the classical CFTP algorithm, consisting of running chains started from all possible states, in parallel. To compare those two algorithms, we need to specify what we mean by *operations*: We say that an algorithm does one operation if it compares two letters of V , to determine if they are equal or not or if the two letters are connected in G . It is intuitively clear, that the two algorithms can be basically decomposed into a sequence of such operations:

- In the CFTP algorithm, the match of the incoming individuals amounts to an investigation of the set of stored compatible items in a determinate order, and thereby, of a sequence of such operations. Second, so does the test of equality of the current states of all Markov chains, at any given time.
- In Algorithm 1, testing the p -condition at all time is again a sequence of operations, and so does the construction of the dynamics of the recursion, from the coalescence time on.

To estimate the number of operations in the two algorithms, for two values of p (3 and 6), we have first drawn realizations of Erdős-Rényi graphs G of parameters (n, q) , that are conditioned to be connected, for various values of the size n and of the connectivity parameter q . We have then tracked the average number of operations for 10 realizations of both algorithms, on the same graph each time. The results are presented in Table 2 and 3.

The results gathered in Tables 2 and 3 tend to indicate that Algorithm 1 is much more efficient than CFTP, and that the performance gap is particular important for sparse graphs. This last fact is an intuitively clear consequence of the fact that the proportion of words satisfying the p -condition is decreasing in the number of edges.

$p = 3$	$q = \frac{1}{8}$	$q = \frac{2}{8}$	$q = \frac{3}{8}$	$q = \frac{4}{8}$	$q = \frac{5}{8}$
$n = 4$, CFTP	2416.7	3282.0	3599.7	3251.1	2046.3
$n = 4$, Algo 1	116.8	397.0	206.8	114.3	153.7
$n = 5$, CFTP	4886.2	3697.8	5002.8	5613.5	5382.1
$n = 5$, Algo 1	78.7	109.8	200.6	128.4	316.8
$n = 6$, CFTP	7594.9	7213.3	5313.6	5782.5	7967.6
$n = 6$, Algo 1	88.3	57.2	86.4	98.8	354.3
$n = 7$, CFTP	10200.0	10168.3	10033.4	9815.9	6503.9
$n = 7$, Algo 1	91.5	97.0	64.5	215.8	969.8
$n = 8$, CFTP	14758.1	14801.3	13651.0	18418.9	9894.3
$n = 8$, Algo 1	43.8	51.1	99.5	472.5	559.7

TABLE 2. Average number of operations of the algorithms for 10 repetitions with $p = 3$ and multiple values of (n, q) .

$p = 6$	$q = \frac{1}{8}$	$q = \frac{2}{8}$	$q = \frac{3}{8}$	$q = \frac{4}{8}$	$q = \frac{5}{8}$
$n = 4$, CFTP	553824.3	404463.5	462846.5	414874.3	340403.4
$n = 4$, Algo 1	6698.2	19479.7	17081.5	19740.5	18955.1
$n = 5$, CFTP	1366869.5	1005914.9	1057218.6	825084.5	764241.6
$n = 5$, Algo 1	7072.9	5165.1	82885.1	20386.7	46694.7
$n = 6$, CFTP	3210451.1	2800719.4	2661488.8	2196083.3	1746776.8
$n = 6$, Algo 1	4480.4	8156.2	6443.1	13585.9	143461.3
$n = 7$, CFTP	5769456.0	7699737.3	5134146.9	4186587.2	1872563.7
$n = 7$, Algo 1	5202.6	5536.9	6117.6	45067.6	591669.1
$n = 8$, CFTP	17509320.0	11375878.3	8355567.8	5383480.2	4598709.4
$n = 8$, Algo 1	1474.8	3570.3	9801.6	55099.9	963611.5

TABLE 3. Average number of operations of the algorithms for 10 repetitions with $p = 6$ and multiple values of (n, q) .

For $q \geq \frac{6}{8}$ however we observe cases where the Algorithm 1 doesn't terminate in a reasonable amount of time.

4.3.5. *Estimating of the loss probability for ML and FCFM.* Algorithm 1 returns a random variable that is distributed from the stationary distribution of the system. This result can be of critical use, to compare the performance of systems, for which no exact characterization of the steady state is known. As an example, we are able to assess the asymptotic loss rate of items of every class. We use this to compare two matching policies in steady state: Match the Longest (ML) and First Come, First Matched (FCFM).

Let $(G = (V, E), \Phi, \mu, p)$ be a discrete matching model with deterministic impatience, and $\tilde{X} = (\tilde{X}_n)_{n \in \mathbb{Z}}$ be the Markov chain of the system. Let π be the stationary distribution for \tilde{X} and for all $(i, j) \in V^2$ such that $(i, j) \notin E$,

$$A_{i,j} = \{x = x_1 \cdots x_j \in \mathbb{X}, x_1 = i \text{ and the arrival is of class } j \text{ in a buffer } x\}.$$

The asymptotic loss rate of items of class i is denoted by

$$(14) \quad \rho(i) := \lim_{N \rightarrow +\infty} \frac{\sum_{n=1}^N \mathbf{1}_{A_n^i}}{N},$$

	ρ	$\rho(1)$	$\rho(2)$	$\rho(3)$	$\rho(4)$	$\rho(5)$
FCFM	0.0293	0.00026	0.00032	0.0132	0.0152	0.00032
ML	0.03122	0.00028	0.0003	0.01536	0.01486	0.00042

TABLE 4. MC Estimates for the asymptotic loss rates for 10^4 repetitions of Algorithm 1 for a random Erdős-Renyi graph of parameters $n = 5, q = 0.6$, for $p = 5$ and μ the uniform distribution.

where for all n ,

$$A_n^i = \{\text{An item of class } i \text{ is lost at time } n\}.$$

An immediate first step analysis implies that

$$(15) \quad \rho(i) = \sum_{j \in V} \pi(A_{i,j}) \nu(j),$$

so $\rho(i)$ can also be interpreted as the probability to lose an item of class i in the system at a given instant, in steady state. Reasoning similarly,

$$\rho = \sum_{i \in V} \rho(i)$$

is the asymptotic loss rate of items (of any class) in the system, and can also be seen as the probability to lose an item (of any class) at a given time, in steady state. Using equation 15 we can then estimate those asymptotic loss rate by running our perfect simulation algorithm 1, and then estimating $\pi(A_{i,j})$ for all $i, j \in V$, by a Monte-Carlo estimate.

Table 4 presents the results over 10^4 simulations, for G a random Erdős-Renyi graph of parameters $n = 5, q = 0.6$, conditioned on being connected, for $p = 5$, and for μ the uniform distribution. Both matching policies FCFM and ML are implemented on the same samples each time. We observe that the overall asymptotic loss rate is slightly, but consistently lower under FCFM than under ML, although nominal loss rates of given nodes can be higher under FCFM.

REFERENCES

- [1] ADAN, I., BUŠIĆ, A., MAIRESSE, J., AND WEISS, G. Reversibility and further properties of FCFS infinite bipartite matching. *Mathematics of Operations Research* 43, 2 (2018), 598–621.
- [2] ADAN, I., KLEINER, I., RIGHTER, R., AND WEISS, G. FCFS parallel service systems and matching models. *Performance Evaluation* 127 (2018), 253–272.
- [3] ADAN, I., AND WEISS, G. Exact FCFS matching rates for two infinite multitype sequences. *Operations Research* 60 (2012), 475–489.
- [4] ANANTHARAM, V., AND KONSTANTOPOULOS, T. A correction and some additional remarks on: stationary solutions of stochastic recursions describing discrete event systems. *Stoch. Process. Appl* 80 (1999), 271–278.
- [5] AVEKLOURIS, A., DEVALVE, L., WARD, A. R., AND WU, X. Matching impatient and heterogeneous demand and supply. *arXiv preprint arXiv:2102.02710* (2021).
- [6] BACCELLI, F., AND BRÉMAUD, P. *Elements of queueing theory: Palm Martingale calculus and stochastic recurrences*, vol. 26. Springer Science & Business Media, 2013.
- [7] BEGEOT, J., MARCOVICI, I., MOYAL, P., AND RAHME, Y. A general stochastic matching model on multigraphs. *arXiv preprint arXiv:2011.05169* (2020).
- [8] BLANCHET, J., AND DONG, J. Perfect sampling for infinite server and loss systems. *Advances in Applied Probability* 47, 3 (2015), 761–786.
- [9] BLANCHET, J., DONG, J., AND PEI, Y. Perfect sampling of gi/gi/c queues. *Queueing Systems* 90, 1 (2018), 1–33.
- [10] BOROVKOV, A., AND FOSS, S. Stochastically recursive sequences and their generalizations. *Siberian advances in Mathematics* 2, 1 (1992), 16–92.
- [11] BOROVKOV, A., AND FOSS, S. Two ergodicity criteria for stochastically recursive sequences. *Acta Applicandae Mathematica* 34, 1 (1994), 125–134.
- [12] BOROVKOV, A. A. *Ergodicity and stability of stochastic processes*. J. Wiley, 1998.

- [13] BUSIC, A., GAUJAL, B., AND VINCENT, J.-M. Perfect simulation and non-monotone markovian systems. In *3rd International Conference Valuetools' 08 (2008)*, ICST.
- [14] BUŠIĆ, A., GUPTA, V., AND MAIRESSE, J. Stability of the bipartite matching model. *Advances in Applied Probability* 45, 2 (2013), 351–378.
- [15] CALDENTEY, R., KAPLAN, E., AND WEISS, G. FCFS infinite bipartite matching of servers and customers. *Adv. Appl. Probab* 41, 3 (2009), 695–730.
- [16] CASTRO, F., NAZERZADEH, H., AND YAN, C. Matching queues with reneging: a product form solution. *Queueing Systems* 96, 3 (2020), 359–385.
- [17] COMTE, C. Stochastic non-bipartite matching models and order-independent loss queues. *Stochastic Models* 38, 1 (2022), 1–36.
- [18] CONNOR, S. B., AND KENDALL, W. S. Perfect simulation for a class of positive recurrent markov chains. *The Annals of Applied Probability* 17, 3 (2007), 781–808.
- [19] FOSS, S. G., AND TWEEDIE, R. L. Perfect simulation and backward coupling. *Stochastic models* 14, 1-2 (1998), 187–203.
- [20] HUBER, M. Perfect sampling using bounding chains. *The Annals of Applied Probability* 14, 2 (2004), 734–753.
- [21] HUBER, M. L. *Perfect simulation*, vol. 148. Monographs on Statistics and Applied Probability, 2016.
- [22] JONCKHEERE, M., MOYAL, P., RAMÍREZ, C., AND SOPRANO-LOTO, N. Generalized max-weight policies in stochastic matching. *arXiv preprint arXiv:2011.04535* (2020).
- [23] KELLY, F. P. Loss networks. *The annals of applied probability* (1991), 319–378.
- [24] KENDALL, W. Geometric ergodicity and perfect simulation. *Electronic Communications in Probability* 9 (2004), 140–151.
- [25] LISEK, B. A method for solving a class of recursive stochastic equations. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete* 60, 2 (1982), 151–161.
- [26] MAIRESSE, J., AND MOYAL, P. Stability of the stochastic matching model. *Journal of Applied Probability* 53, 4 (2016), 1064–1077.
- [27] MOYAL, P. Stability of a processor-sharing queue with varying throughput. *Journal of Applied Probability* 45, 4 (2008), 953–962.
- [28] MOYAL, P. On queues with impatience: stability, and the optimality of earliest deadline first. *Queueing Systems* 75, 2 (2013), 211–242.
- [29] MOYAL, P. A generalized backward scheme for solving non-monotonic stochastic recursions. *The Annals of Applied Probability* 25, 2 (2015), 582–599.
- [30] MOYAL, P., BUŠIĆ, A., AND MAIRESSE, J. A product form for the general stochastic matching model. *Journal of Applied Probability* 58, 2 (2021), 449–468.
- [31] NAZARI, M., AND STOLYAR, A. L. Reward maximization in general dynamic matching systems. *Queueing Systems* 91, 1 (2019), 143–170.
- [32] PROPP, J. G., AND WILSON, D. B. Exact sampling with coupled markov chains and applications to statistical mechanics. *Random Structures & Algorithms* 9, 1-2 (1996), 223–252.
- [33] RAHME, Y., AND MOYAL, P. A stochastic matching model on hypergraphs. *Advances in Applied Probability* 53, 4 (2021), 951–980.