Discrete Mathematics and Theoretical Computer Science **AC**, 2003, 69–82

Joint Burke's Theorem and RSK Representation for a Queue and a Store

Moez Draief[1] and Jean Mairesse[1] and Neil O'Connell[2]

[1]*LIAFA, Univ. Paris 7, case 7014, 2 pl. Jussieu, 75251 Paris 05, France.* draief,mairesse@liafa.jussieu.fr
[2]*Mathematics Institute, Univ. Warwick, Coventry CV4 7AL, United Kingdom.* noc@maths.warwick.ac.uk

Consider the single server queue with an infinite buffer and a FIFO discipline, either of type $M/M/1$ or Geom/Geom/1. Denote by $\mathcal{A}$ the arrival process and by s the services. Assume the stability condition to be satisfied. Denote by $\mathcal{D}$ the departure process in equilibrium and by r the time spent by the customers at the very back of the queue. We prove that $(\mathcal{D},r)$ has the same law as $(\mathcal{A},s)$ which is an extension of the classical Burke Theorem. In fact, r can be viewed as the departures from a dual *storage* model. This duality between the two models also appears when studying the transient behavior of a tandem by means of the RSK algorithm: the first and last row of the resulting semi-standard Young tableau are respectively the last instant of departure in the queue and the total number of departures in the store.

Keywords: Single server queue, storage model, Burke's theorem, non-colliding random walks, tandem of queues, Robinson-Schensted-Knuth algorithm

1 Introduction

The main purpose of this paper is to clarify the interplay between two models of queueing theory. The first model is the very classical single server queue with an infinite buffer and a FIFO discipline. The second, less common but very natural, model can be described as a queue operating in slotted time with batch arrivals and services. It was studied for instance in [3]. To clearly distinguish between the two models, we choose to describe the second one with a different terminology and as a storage model.

We prove first that the two models are linked in a very strong way. We set up an abstract model with an ordered pair of input variables $(\mathcal{A},s)$ and an ordered pair of output variables $(\mathcal{D},r) = \Phi(\mathcal{A},s) = (\Phi_1(\mathcal{A},s),\Phi_2(\mathcal{A},s))$. On the one hand, the queueing model corresponds to $\mathcal{D} = \Phi_1(\mathcal{A},s)$ with $\mathcal{A}$ and s being respectively the arrivals and services and $\mathcal{D}$ the departures. On the other hand, the storage model corresponds to $r = \Phi_2(\mathcal{A},s)$, with s and $\mathcal{A}$ being respectively the supplies (arrivals) and requests (services) and r the departures. The interpretation of either r for the queue or $\mathcal{D}$ for the store is much less natural.

Then we assume that the random variables driving the dynamic are either exponentially or geometrically distributed, and we consider the models in equilibrium (under the stability condition). In this situation, it is well known that a Burke's type Theorem holds: the departures and the arrivals have the same law [5, 19, 3]. This can be considered as one of the cornerstones of queueing theory, see for instance the books [4, 10, 20] for discussions and related materials. Here we prove a 'joint' version of the result:

$(\mathcal{D},r)$ has the same law as $(\mathcal{A},s)$. This joint Burke Theorem is new, although it is similar in spirit to the results proved in [17, 12] for a variant: queues with *unused services*. Also, in the geometric case, we use an original method of proof based on the reversibility of a symmetrized version of the workload process (instead of the queue length process). As in [17, 12], the joint Burke's Theorem can be used to obtain a representation theorem for the joint law of two independent random walks with exponential or geometric jumps conditioned on never colliding.

A second facet of the duality between queues and stores appears when considering the Robinson–Schensted–Knuth (RSK) algorithm. Consider K queues (stores) in tandem: the departures from a queue (store) are the arrivals (supplies) at the next one. Initially, the network is empty except for an infinite number of customers (infinite supply) in the first queue (store).

Here we assume that the variables driving the dynamic are $\mathbb{N}$-valued r.v.'s without any assumption on their joint distribution. Building on ideas developed in [2, 15], we can study the transient evolution as follows. Consider the family of r.v.'s $(u(i,j))_{1\leq i\leq N, 1\leq j\leq K}$ where $u(i,j)$ is the service of the i-th customer at queue j, resp. the request at time slot i in store $K+1-j$. Apply the RSK algorithm, see [11, 21, 22], to this family and let P be the resulting semi-standard Young tableau (here we do not consider the recording tableau Q). Let $\lambda_1 \geq \cdots \geq \lambda_K \geq 0$ be the lengths of the successive rows of P. Classically [11], we have $\lambda_1 = \max_{\pi\in\Pi}\sum_{(i,j)\in\pi} u(i,j)$, where Π is the set of paths in $\mathbb{N}^2$ going from $(1,1)$ to (N,K) and which are increasing and consist of adjacent points. Moreover, it is well known in queueing theory [13, 23, 7] that $\max_{\pi\in\Pi}\sum_{(i,j)\in\pi} u(i,j)$ is D: the instant of departure of customer N from queue K. Pasting the two results together gives the folklore observation that $\lambda_1 = D$. Here we complete the picture by proving that $\lambda_K = R$, where R is the total number of departures from the last store in the tandem up to time slot N. Again, this identity is proved in two steps by showing that $\lambda_K = \min_{\pi\in\widetilde{\Pi}}\sum_{(i,j)\in\pi} u(i,j) = R$, where $\widetilde{\Pi}$ is a different set of paths in the lattice. To summarize, we obtain on the same Young tableau the total departures for the two tandem models.

2 Notations

We work on a probability space $(\Omega,\mathcal{F},P)$. The indicator function of an event $\mathcal{A}\in\mathcal{F}$ is denoted by $\mathbb{1}_{\mathcal{A}}$. We use the symbol $\sim$ to denote the equality in distribution of random variables. Depending on the context, $|A|$ is the cardinal of set A or the length of word A. We set $\mathbb{N}^* = \mathbb{N}\setminus\{0\}$ and $x^+ = x\vee 0 = \max(x,0)$. We use the convention that $\sum_{i=j}^k u_i = 0$ when $j>k$.

Below, a *point process* is a stochastic simple point process on $\mathbb{R}$ with an infinite number of positive and negative points. We identify a point process with the random ordered sequence of its points: $\mathcal{A} = (A_n)_{n\in\mathbb{Z}}$ with $A_n < A_{n+1}$ for all n. Observe that the numbering of the points is defined up to a translation in the indices. For any interval I, we define the *counting* random variable: $\mathcal{A}(I) = \sum_{n\in\mathbb{Z}}\mathbb{1}_{\{A_n\in I\}}$.

A *marked point process* is a couple $(\mathcal{A},c) = (A_n,c_n)_{n\in\mathbb{Z}}$ where $\mathcal{A} = (A_n)_{n\in\mathbb{Z}}$ is a point process and $c = (c_n)_{n\in\mathbb{Z}}$ is a sequence of r.v.'s valued in some state space. The *mark* c_n is *associated* to the point A_n of the point process. For precisions concerning point processes see [6].

Given a point process $\mathcal{A} = (A_n)_{n\in\mathbb{Z}}$, the *reversed* point process $\mathcal{R}(\mathcal{A})$ is the point process obtained by reversing the direction of time; i.e. $\mathcal{R}(\mathcal{A}) = (-A_{-n})_{n\in\mathbb{Z}}$. Given a marked point process $(\mathcal{A},c) = (A_n,c_n)_{n\in\mathbb{Z}}$, the *reversed* marked point process is $\mathcal{R}(\mathcal{A},c) = (-A_{-n},c_{-n})_{n\in\mathbb{Z}}$.

Given a càdlàg, i.e. right-continuous and left-limited, random process $Y = (Y(t))_{t\in\mathbb{R}}$ valued in $\mathbb{R}$, we define the *reversed* process $\mathcal{R}\circ Y = (\mathcal{R}\circ Y(t))_{t\in\mathbb{R}}$ as the càdlàg modification of the process $(Y(-t))_{t\in\mathbb{R}}$.

Denote by $\mathcal{N}_+(Y)$ and $\mathcal{N}_-(Y)$ the point processes (with a possibly finite number of points) corresponding respectively to the positive and negative jumps of Y, that is for any interval I of $\mathbb{R}$,

$$\mathcal{N}_+(Y)(I) = \int_I \mathbb{1}_{\{Y(u)>Y(u-)\}} du, \ \mathcal{N}_-(Y)(I) = \int_I \mathbb{1}_{\{Y(u)<Y(u-)\}} du \,. \tag{1}$$

3 The Model

Let $\mathcal{A} = (A_n)_{n\in\mathbb{Z}}$ be a point process and assume that $A_0 \leq 0 < A_1$. We define the $\mathbb{R}_+^*$-valued sequence of r.v's $a = (a_n)_{n\in\mathbb{Z}}$ by $a_n = A_{n+1} - A_n$. Let $s = (s_n)_{n\in\mathbb{Z}}$ be another $\mathbb{R}_+^*$-valued sequence of r.v's. The marked point process $(\mathcal{A}, s)$ is the *input* of the model.

Define the sequence of r.v.'s $\mathcal{D} = (D_n)_{n\in\mathbb{Z}}$ by

$$D_n = \sup_{k\leq n} \left[A_k + \sum_{i=k}^{n} s_i\right] . \tag{2}$$

A priori the D_n's are valued in $\mathbb{R}\cup\{+\infty\}$. Assume from now on that $(\mathcal{A}, s)$ is such that the D_n's are almost surely finite. They satisfy the recursive equations:

$$D_{n+1} = \max(D_n, A_{n+1}) + s_{n+1} \,. \tag{3}$$

We have $\forall n$, $D_n < D_{n+1}$. Set $d_n = D_{n+1} - D_n$. We define an additional sequence of r.v.'s $r = (r_n)_{n\in\mathbb{Z}}$, valued in $\mathbb{R}_+^*$, by

$$r_n \quad = \quad \min(D_n, A_{n+1}) - A_n \,. \tag{4}$$

The marked point process $(\mathcal{D}, r)$ is the *output* of the model. By summing (3) and (4), we get the following interesting relation between input and output variables

$$r_n + d_n = a_n + s_{n+1} \,. \tag{5}$$

In view of the future analysis, it is convenient to define the following auxiliary variables. Let $w = (w_n)_{n\in\mathbb{Z}}$ be the sequence of r.v.'s valued in $\mathbb{R}_+$ and defined by

$$w_n = D_n - s_n - A_n = \sup_{k\leq n-1} \left[\sum_{i=k}^{n-1} (s_i - a_i)\right]^+ . \tag{6}$$

These r.v.'s satisfy the recursive equations:

$$w_{n+1} = [w_n + s_n - a_n]^+ \,. \tag{7}$$

Using the variable w_n, we can give alternative definitions of D_n and r_n:

$$\forall l \leq n, \ D_n = \left[w_l + A_l + \sum_{i=l}^{n} s_i\right] \vee \max_{l<k\leq n} \left[A_k + \sum_{i=k}^{n} s_i\right] , \tag{8}$$

$$r_n = \min\{w_n + s_n, a_n\} \,. \tag{9}$$

At last, we define the càdlàg random process $Q = (Q(t))_{t\in\mathbb{R}}$, valued in $\mathbb{N}$, by

$$Q(t) = \sum_{n\in\mathbb{Z}} \mathbb{1}_{\{A_n \leq t < D_n\}} \,. \tag{10}$$

Lemma 3.1. *We have* $\mathcal{N}_+(Q) = \mathcal{A}$ *and* $\mathcal{N}_-(Q) = \mathcal{D}$*. Furthermore,* $D_{-Q(0)} \leq 0 < D_{-Q(0)+1}$.

The proof is straightforward. We now interpret the variables defined above in two different contexts: a queueing model and a storage model.

3.1 The single-server queue

A bi-infinite string of customers is served at a queueing facility with a single server. Each customer is characterized by an instant of arrival in the queue and a service demand. Customers are served upon arrival in the queue and in their order of arrival. Since there is a single server, a customer may have to wait in a buffer before the beginning of its service. Using Kendall's nomenclature, our model is a $./ ./1/\infty/$FIFO queue.

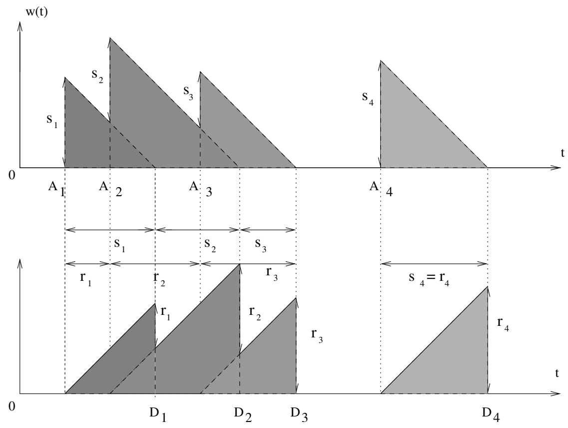

Fig. 1: The dual variables $(s_n)_n$ and $(r_n)_n$

The customers are numbered by $\mathbb{Z}$ according to their order of arrival in the queue. Let A_n be the instant of **A**rrival of customer n and s_n its **S**ervice time. Then the variables defined in (2)-(10) have the following interpretations:

- D_n is the instant of departure of customer n from the queue, after completion of its service;
- w_n is the waiting time of customer n in the buffer between its arrival and the beginning of its service;
- $Q(t)$ is the number of customers in the queue at instant t (either in the buffer or in service); $Q = (Q(t))_t$ is called the *queue-length* process;
- r_n is the time spent by customer n at the *very back* of the queue.

The variables $(r_n)_n$ are less classical in queueing theory although they have already been considered [18]. They should be viewed as being dual to the services $(s_n)_n$ as illustrated in Figure 1. On the upper part of the figure, we have represented the *workload* process $(W(t))_t$, where $W(t)$ is the waiting time of a virtual customer arriving at instant t (see (11) for the formal definition).

3.2 The storage model

Some product P is supplied, sold and stocked in a store in the following way. Events occur at integer-valued epochs, called *slots*. At each slot, an amount of P is supplied and an amount of P is asked for by potential buyers. The rule is to meet all the demand, if possible. The demand of a given slot which is not met *is lost*. The supply of a given slot which is not sold *is not lost* and is stocked for future consideration.

Let s_n be the amount of P **S**upplied at slot $n+1$, and let a_n be the amount of P **A**sked for at the same slot. The variables in (2)-(10) can be interpreted in this context:

- w_n is the level of the stock at the end of slot n. It evolves according to (7);
- r_n is the demand met at slot $n+1$, see equation (9); it is the amount of P departing at slot $n+1$;

The variables $(D_n)_n$ and the function Q do not have a natural interpretation in this model.

The evolution of the store is summarized in Figure 2. The indices may seem weird (s_n, a_n, r_n, for time slot $n+1$). They were chosen that way to get better looking formulas in §5.

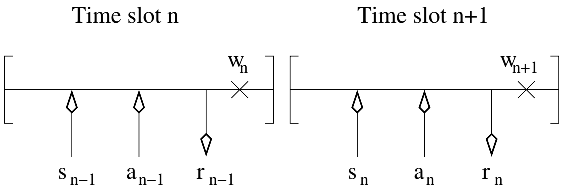

Fig. 2: The storage model

It is important to remark that while the equations driving the *single server queue* and the *storage model* are exactly the same, it is not the same variables that make sense in the two models. The important variables are the ones corresponding to the departures from the system. The departures are coded in the variables $(D_n)_n$ for the single server queue and in the variables $(r_n)_n$ for the storage model. On the other hand, interpreting the variables $(r_n)_n$ in the single server queue or the variables $(D_n)_n$ in the storage model is not so immediate.

Observe that it would be possible to describe the above storage model with a "queueing" terminology (a queue with slotted time, batch arrivals (s_n) and batch services (a_n)). It is the description used in [3] for instance. We have avoided it on purpose to clearly separate the models of §3.1 and §3.2 and therefore minimize the possibility of confusion.

4 Equilibrium Behavior: Around Burke's Output Theorem

We consider the exponential or geometric version of our model in equilibrium. We prove a Burke's type result: $(\mathcal{D}, r) \sim (\mathcal{A}, s)$. The relevant result for the sequence of departures in one of the two models will then follow by forgetting one of the two variables. A discussion of the literature and of the increment of the present version is carried out in §4.3.

4.1 Output theorem in the exponential case

Let $\mathcal{A}$ be an homogeneous Poisson process of intensity $\lambda \in \mathbb{R}_+^*$. We set $\mathcal{A} = (A_n)_{n\in\mathbb{Z}}$ with $A_0 < 0 < A_1$. Recall that $a_n = A_{n+1} - A_n$. Then $(a_n)_{n\geq 1}$ is a sequence of i.i.d. r.v.'s with exponential distribution of parameter λ. Let $s = (s_n)_{n\in\mathbb{Z}}$ be a sequence of i.i.d. random variables, independent of $\mathcal{A}$, with exponential distribution of parameter μ. We assume that $\lambda < \mu$.

We now consider the marked point process $(\mathcal{A}, s)$ as being the input of the model of Section 3. The sequence $(w_n)_n$ is a random walk valued in $\mathbb{R}_+$ with an absorbing barrier at 0. Under the *stability* condition $\lambda < \mu$, this random walk has a negative drift. It implies that the random variables D_n defined in (2) are indeed almost surely finite. We have the following result.

Theorem 1. *The marked point process $(\mathcal{D}, r)$ has the same law as the marked point process $(\mathcal{A}, s)$.*

Proof. The principle is the same as in Reich's proof [19]. The only subtlety is to keep track of the indices to make sure that the variable r_n is the mark of the point D_n. Here is a sketch of the argument.

Knowing Q, one can recover the arrivals and services: $(A_n, s_n)_n = \varphi(Q)$. Playing with Equations (2)-(4), we can prove that $\varphi \circ \mathcal{R}(Q) = (-D_{-n+Q(0)+1}, r_{-n+Q(0)+1})_n = \mathcal{R}(\mathcal{D}, r)$. (In particular, the negative, resp. positive, jumps of the reversed process $\mathcal{R}(Q)$. correspond to the positive, resp. negative, jumps of the process Q; and we have Lemma 3.1.) Now, the queue-length process Q is a stationary birth-and-death process, hence reversible: $\mathcal{R}(Q) \sim Q$. It implies that $(\mathcal{A}, s) \sim \mathcal{R}(\mathcal{D}, r)$. Furthermore, we have clearly $\mathcal{R}(\mathcal{D}, r) \sim (\mathcal{D}, r)$ since $\mathcal{R}(\mathcal{D}, r)$ is a homogeneous Poisson process marked with an i.i.d. sequence of r.v.'s. It concludes the proof. □

Corollary 4.1. *In the queueing model, the departure process $\mathcal{D}$ is a Poisson process of intensity λ. In the storage model, the sequence $(r_n)_n$ of the amounts of product P departing at successive slots, is a sequence of i.i.d. exponential r.v.'s of parameter μ.*

4.2 Output theorem in the geometric case

Let $\mathcal{A}$ be a Bernoulli point process of parameter $p \in (0,1)$, that is: all the points are integer valued, there is a point at a given integer with probability p, and the presence of points at different integers are independent. As before, set $\mathcal{A} = (A_n)_n$ with $A_0 \leq 0 < A_1$ and $a_n = A_{n+1} - A_n$. Then the sequence $(a_n)_{n\geq 1}$ is a sequence of i.i.d. geometric r.v.'s with parameter p ($\forall k \in \mathbb{N}^*$, $P\{a_1 = k\} = (1-p)^{k-1}p$). Let $(s_n)_n$ be a sequence of i.i.d. geometric r.v.'s with parameter $q \in (0,1)$, and independent of $\mathcal{A}$. We assume that $p < q$ (stability condition).

Let the marked point process $(\mathcal{A}, s)$ be the input of the model of Section 3. As in §4.1, the model is stable and the output $(\mathcal{D}, r)$ is a marked point process. Define

$$W(t) = \left[w_n + s_n - (t - A_n)\right]^+, \quad \text{for } t \in [A_n, A_{n+1})\,, \tag{11}$$

where $(w_n)_n$ is defined in (6). Observe that $W(A_n^-) = w_n$. For the queueing model, $W(t)$ is the total amount of service remaining to be done by the server at instant t, and $W = (W(t))_t$ is called the *workload* process. Define for all $n \in \mathbb{Z}$:

$$B_n = C_{n-1} + a_{n-1}, \;\; C_n = B_n + s_n\,. \tag{12}$$

Given $(a_n)_{n\in\mathbb{Z}}$ and $(s_n)_{n\in\mathbb{Z}}$, the above recursions enable to define $(B_n)_{n\in\mathbb{Z}}$ and $(C_n)_{n\in\mathbb{Z}}$ knowing C_0. Let us set $C_0 = A_0$. The intervals $[B_n, C_n)$ and $[C_n, B_{n+1})$ partition $\mathbb{R}$. Define the *(reflected) zigzag process* $Z = (Z(t))_{t\in\mathbb{R}}$ as follows:

$$Z(C_0) = W(A_0^+), \qquad Z(t) = \begin{cases} Z(B_n) + (t - B_n) & \text{for } t \in [B_n, C_n) \\ \big[Z(C_n) - (t - C_n)\big]^+ & \text{for } t \in [C_n, B_{n+1}) \end{cases}. \tag{13}$$

On an interval of type $[B_n, C_n)$, we have $dZ/dt = 1$ and on an interval of type $[C_n, B_{n+1})$, we have $dZ/dt = (-1)\mathbb{1}_{Z>0}$. The zigzag process is a symmetrization of the workload process W. We have represented in Figure 3 a trajectory of W and the corresponding trajectory of Z. We have the following result:

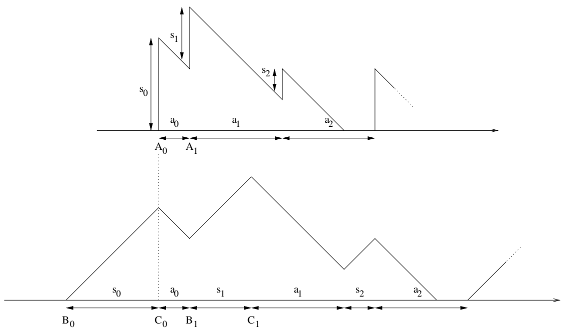

Fig. 3: The workload process (top) and the zigzag process (bottom)

Theorem 2. *The marked point process $(\mathcal{D}, r)$ has the same law as the marked point process $(\mathcal{A}, s)$.*

Proof. The proof is similar to the one of Theorem 1 with Z playing the role of Q. Indeed, focusing on Figure 3, one can see that the stationary version of Z is a reversible process. The formal proof is given in [8, Theorem 2]. Also it is clear that $(\mathcal{A}, s)$ can be obtained from Z by applying some operator ψ. Applying the same operator to $\mathcal{R}(Z)$ yields $\mathcal{R}(\mathcal{D}, r)$. □

The zigzag process is clearly also reversible in the exponential model. Hence the proof used for Theorem 2 can also be used to get Theorem 1.

4.3 Comments on the different proofs of Burke Theorem

Reflecting on the above, there are three different ways to prove Burke Theorem, be it in the exponential or geometric case. The first way is by using analytic methods, the second is by using the reversibility of the queue length process Q, and the third is by using the reversibility of the zigzag process Z.

The original proof of Burke is for the exponential model using analytic methods [5]. For the geometric model, an analytic proof was given by Azizoğlu and Bedekar [3]. For the exponential model, the idea of using the reversibility of Q to get the result is due to Reich [19]. This proof has become a cornerstone of queueing theory, it has been extended to various contexts and has given birth to the concept of *product form networks* [4, 10]. Reich's proof does not translate directly to the geometric model. Of course,

$(Q(n))_{n\in\mathbb{Z}}$ is a reversible birth-and-death Markov chain. However, a difficulty arises: It is not possible to reconstruct $\mathcal{A}$ and s from Q. Indeed, on the event $\{Q(n-1)=Q(n)>0\}$, two cases may occur: there is either no departure and no arrival at instant n, or one departure and one arrival; and it is not possible to distinguish between them knowing only Q. One feasible solution is to add an auxiliary sequence that contains the lacking information but the details become quite intricate. This program has been carried out in [12, Theorem 4.1] for a variant: the geometric model with *unused services*. The above idea of using the zigzag process to prove Burke Theorem is original. This zigzag process was studied, with a different motivation, in [8].

In the original references [5, 19] and in all the classical textbooks presenting the result [4, 10, 20], the version proved is: $\mathcal{A}\sim\mathcal{D}$: ("Poisson Input Poisson Output"). In [3], the result proved is $s\sim r$. The complete version $(\mathcal{A},s)\sim(\mathcal{D},r)$ appears first in [17, Theorem 3] for a variant: the exponential model with *unused services*. This is extended to the geometric model with *unused services* in [12]. Brownian analogues are proved in [9, 16].

4.4 Non-colliding random walks

Following the lines of thought in [17, 12, 8], it is possible to use Theorems 1 and 2 to get representation results for non-colliding random walks.

Proposition 4.2. *Let the sequences of r.v.'s $(a_n)_{n\in\mathbb{N}^*}$ and $(s_n)_{n\in\mathbb{N}^*}$ be as in §4.1 or as in §4.2. The conditional law of $(\sum_{i=1}^n a_i,\ \sum_{i=1}^n s_i)$, given that $\{\sum_{i=1}^k a_i \geq \sum_{i=1}^{k+1} s_i,\ \forall k\geq 0\}$ is the same as the unconditional law of $(\max_{1\leq j\leq n}\{\sum_{i=1}^j a_i+\sum_{i=j+1}^{n+1} s_i\},\ \min_{1\leq j\leq n}\{\sum_{i=2}^j s_i+\sum_{i=j+1}^n a_i\})$.*

It is possible to extend Proposition 4.2 to the limiting case $E[a_1]=E[s_1]$, and also to higher dimensions, by adapting the methods of [17, 12, 8] to the present setting.

5 Transient Behavior and RSK Representation

5.1 The saturated tandem

We consider another aspect of the dynamic of queues and stores: the transient evolution for the model starting empty. More precisely, consider the model of §3 under the assumption that $w_0=A_0=s_0=0$ (which implies $D_0=r_0=0$) and focus on the customers, resp. time slots, from 1 onwards.

It is convenient to describe such a model with a different perspective. We first do it for the queue. View the arrivals as being the departures from a *virtual queue* having at instant 0 an infinite number of customers (labelled by $\mathbb{N}^*$) in its buffer. The service time of customer n in the virtual queue is a_{n-1}. We describe this as a *saturated tandem of two queues*.

Let us turn our attention to the store. View the supplies as being the departures from a virtual store having an infinite stock at the end of time slot 0. In the virtual store, the request (=departure) at time slot n is s_n. This is a *saturated tandem of two stores*.

Now we want to fit these two descriptions together. Denote the virtual queue/store as *queue/store 1* and the other one as *queue/store 2*. For convenience, set $u(n,1)=a_{n-1}$ and $u(n,2)=s_n$ for all $n\geq 1$. The saturated tandem is completely specified by the family $(u(n,i),n\in\mathbb{N}^*,i=1,2)$ of input variables. These variables are the services, resp. requests, when the model is seen as a tandem of queues, resp. stores. Next table gives the input variables in the saturated tandems of two queues/stores.

Customer / Time slot	1	2	3	$\cdots$	n
Queue 2 / (Virtual) Store 1	$u(1,2)=s_1$	$u(2,2)=s_2$	$u(3,2)=s_3$		$u(n,2)=s_n$
(Virtual) Queue 1 / Store 2	$u(1,1)=a_0$	$u(2,1)=a_1$	$u(3,1)=a_2$		$u(n,1)=a_{n-1}$

A couple of observations are in order. Observe that $u(n,i)$ is the service of customer n in queue i, and the request at time slot n in store $3-i$. In other words, the elements (queues or stores) associated with a given sequence $(u(\cdot,i)), i=1,2$, are crossed in reverse orders in the queueing/storage tandem. This is illustrated in Figure 6 (set $K=2$). Observe also that there is a shift in the time slots for the storage model: the departure from store 1 at time slot n is the supply of store 2 at time slot $n+1$ (contrast this with the situation for the queues). This is coherent with a model in which we view a time slot as being decomposable in three consecutive stages: first, the supply arrives; second, the request is made; third, the departure occurs. See Figure 2.

The above setting is naturally extended to define the *saturated tandem of K queues/stores*. Such a model is entirely defined by a family of $\mathbb{N}$-valued† r.v.'s $(u(i,j))_{i\in\mathbb{N}^*,j\in\{1,\dots,K\}}$. For the queueing model: (i) at instant 0, queue 1 has an infinite number of customers labelled by $\mathbb{N}^*$ in its buffer, and the other queues are empty; (ii) $u(i,j)$ is the service of customer i at queue j; (iii) the instant of departure of customer n from queue i is the instant of arrival of customer n in queue $i+1$. For the storage model: (i) at the end of time slot 0, store 1 has an infinite stock and the other stores have an empty stock; (ii) $u(i,K+1-j)$ is the request at time slot i in store j; (iii) the departure at time slot n from store i is the supply at time slot $n+1$ in store $i+1$.

The models are depicted in Figure 6.

5.2 Robinson–Schensted–Knuth representation

A *partition* of $n\in\mathbb{N}^*$ is a sequence of integers $\lambda=(\lambda_1,\dots,\lambda_k)$ such that $\lambda_1\geq\cdots\geq\lambda_k\geq 0$ and $\lambda_1+\cdots+\lambda_k=n$. We use the notation $\lambda\vdash n$. By convention, we identify partitions having the same non-zero components. The *(Ferrers) diagram* of $(\lambda_1,\dots,\lambda_k)\vdash n$ is a collection of n boxes arranged in left-justified rows, the i-th row starting from the top consisting of λ_i boxes. A *(semi-standard Young) tableau* on the alphabet $\{1,\dots,k\}$ is a diagram in which each box is filled in by a label from $\{1,\dots,k\}$ in such a way that the entries are weakly increasing from left to right along the rows and strictly increasing down the columns. The *shape* of a diagram or tableau is the underlying partition. A *standard tableau* of size n is a tableau of shape $\lambda\vdash n$ whose entries are from $\{1,\dots,n\}$ and are distinct. In Figure 4 we have represented a diagram on the left and a tableau of the same shape on the right.

The Robinson–Schensted–Knuth row-insertion algorithm (RSK algorithm) takes a tableau T and $i\in\mathbb{N}^*$ and constructs a new tableau $T\leftarrow i$. The tableau $T\leftarrow i$ has one more box than T and is constructed as follows. If i is at least as large as the labels of the first (upper) row of T, add a box labelled i to the end of the first row of T and stop the procedure. Otherwise, find the leftmost entry in the first row which is strictly larger than i, relabel the corresponding box by i and apply the same procedure recursively to the second row and to the bumped label. By convention, an empty row has label 0. With this convention, the above procedure stops.

Consider a word $v=v_1\cdots v_n$ over the alphabet $\{1,\dots,k\}$. The tableau associated with v is by definition

$$P=(\cdots((T_0\leftarrow v_1)\leftarrow v_2)\cdots\leftarrow v_n),$$

† In §3, the r.v.'s were valued in $\mathbb{N}^*$. This restriction is not necessary here.

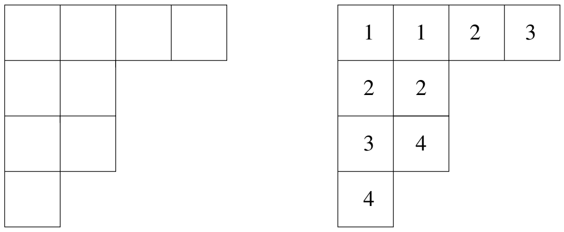

Fig. 4: Ferrers diagram and semi-standard Young tableau of shape $(4,2,2,1) \vdash 9$

where T_0 is the empty tableau. Observe that P has at most k non-empty rows. Classically, the length of the top (and longest) row of P is equal to the longest weakly-increasing subsequence in v.

Remark 5.1. While building the tableau P, it is possible to build another tableau of the same shape in which the entries, labelled from 1 to $|v|$, record the order in which the boxes are added. This *recording tableau* is a standard tableau of size $|v|$. By doing this, one defines a bijection between words of $\{1,\ldots,k\}^n$ and ordered pairs of tableaux of the same shape, the first being semi-standard over the alphabet $\{1,\ldots,k\}$ and the second being standard and of size n. Here, we do not need this result and we do not consider the recording tableau.

Consider a family $U = (u(i,j))_{(i,j)\in\{1,\ldots,N\}\times\{1,\ldots,K\}}$ of r.v.'s valued in $\mathbb{N}$. (Here we do not make any assumption on the distribution of these r.v.'s.) We associate with U the word over the alphabet $\{1,\ldots,K\}$ defined by $w(U) = w_1 \ldots w_N$ and

$$w_i = \underbrace{1\cdots 1}_{u(i,1)} \; \underbrace{2\cdots 2}_{u(i,2)} \; \underset{\cdots}{\cdots} \; \underbrace{K\cdots K}_{u(i,K)} \; . \tag{14}$$

Set $M = |w(U)| = \sum_{i=1}^{N}\sum_{j=1}^{K} u(i,j)$. For $i = 1,\ldots,K$, define

$$\forall n \leq M, \;\; x_i(n) = |\{j \leq n \,|\, w(U)_j = i\}| \, .$$

Given two maps $x,y : \{1,\ldots,N\} \longrightarrow \mathbb{N}$ ($N \in \mathbb{N}^*$), define the maps $x \triangledown y, x \vartriangle y : \{1,\ldots,N\} \longrightarrow \mathbb{N}$ as follows:

$$\forall n \leq N,\; x\triangledown y(n) = \max_{0\leq m\leq n}\Big[x(m)+y(n)-y(m)\Big], \;\; x \vartriangle y(n) = \min_{0\leq m\leq n}\Big[x(m)+y(n)-y(m)\Big] \, . \tag{15}$$

Denote by $P(U)$ the tableau obtained from $w(U)$ by applying the RSK algorithm and let $(\lambda_1,\ldots,\lambda_K)$ be its shape. The following holds

$$\lambda_1 = x_1 \triangledown x_2 \triangledown \ldots \triangledown x_K(M), \;\; \lambda_K = x_K \vartriangle \ldots \vartriangle x_2 \vartriangle x_1(M) \, , \tag{16}$$

where the operations $\triangledown$, $\vartriangle$ are performed from left to right (the operations $\triangledown$, $\vartriangle$ are non-associative). The expression for λ_1 follows from the fact that λ_1 is the longest weakly-increasing subsequence in $w(U)$. The expression for λ_K is proved in [15, Theorem 3.1]. In fact, the result from [15] is more general: there exists a min-max-type operator Γ_K such that $(\lambda_1,\ldots,\lambda_K) = \Gamma_K(x_1,\ldots,x_K)$.

A *lattice path* is a sequence $\pi = ((i_1,j_1),\ldots,(i_l,j_l))$ with $(i_k,j_k) \in \mathbb{Z}^2$. The *steps* of π are the differences $(i_{k+1}-i_k, j_{k+1}-j_k)$, $k = 1,\ldots l-1$.

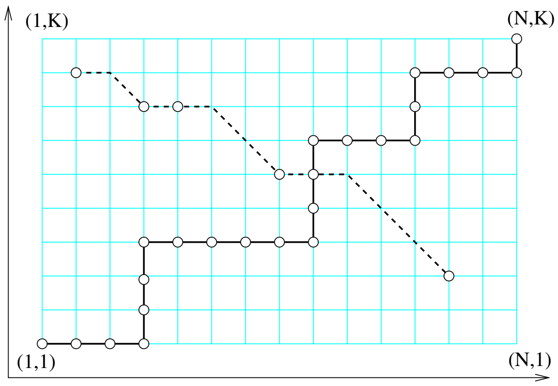

Fig. 5: A path from Π (plain line) and a path from $\widetilde{\Pi}$ (dashed line)

Let Π be the set of lattice paths from $(1,1)$ to (N,K) with steps of the type $(1,0)$ or $(0,1)$. All the paths in Π have the same number of nodes: $N+K-1$. Let $\widetilde{\Pi}$ be the set of lattice paths from one of the points in $\{(1+i,K-i), i=0,\ldots,K-1\}$ to one of the points in $\{(N-j,1+j), j=0,\ldots,N-1\}$ with steps of the type $(1+i,-i)$, $i\geq 0$. Observe that all the paths in $\widetilde{\Pi}$ also have the same number of nodes: $(N-K+1)^+$. In particular, $\widetilde{\Pi}$ is empty when $N<K$. An example of a path from each of the two sets is provided in Figure 5.

The following result holds. When interpreting it, recall that queues and stores are arranged in opposite orders in the tandems, see Figure 6. It is also fruitful to compare the statements of Proposition 4.2 and Theorem 3 for $K=2$.

Theorem 3. *Consider a saturated tandem of K queues/stores with variables $u(i,j), i\in\mathbb{N}^*, j\in\{1,\ldots,K\}$, as defined in §5.1. Fix $N\in\mathbb{N}^*$. Let D be the instant of departure of customer N from queue K and let R be the cumulative departures over the time slots 1 to N in store K. Define $w(U)$ as in (14). Let $(\lambda_1,\ldots,\lambda_K)$ be the shape of the tableau associated with $w(U)$. We have*

$$\lambda_1 = \max_{\pi\in\Pi}\Big[\sum_{(i,j)\in\pi} u(i,j)\Big] = D \tag{17}$$

$$\lambda_K = \min_{\pi\in\widetilde{\Pi}}\Big[\sum_{(i,j)\in\pi} u(i,j)\Big] = R\,. \tag{18}$$

Proof. The result for λ_1 is essentially due to Schensted. The left equality in (17) is (16) and the right equality appears for instance in [13, 23, 7]. As far as λ_K is concerned, the left equality can be obtained by direct inspection starting from (16). The right equality can be recovered by playing with the equations (2)-(4).

We conclude by providing the complete proof of (18), assuming (17), in the case $K=2$ where the argument is simpler. We use the notations of §3. Using (9) and (7), we get that $r_i = s_i - w_{i+1} + w_i$. Accordingly, the cumulative departures over the time slots 1 to N are: $R_N = \sum_{i=0}^{N-1} r_i = \sum_{i=1}^{N-1} s_i - w_N$.

Developing (7) and using that $w_1 = 0$, we obtain $w_N = \max_{1\le j\le N-1}[\sum_{i=j}^{N-1} s_i - a_i]^+$. Hence we have

$$\begin{aligned} R_N &= \sum_{i=1}^{N-1} s_i - \max_{1\le j\le N-1}\left[\sum_{i=j}^{N-1} s_i - a_i\right]^+ \\ &= \min_{1\le j\le N}\left[\sum_{i=1}^{j-1} s_i + \sum_{i=j}^{N-1} a_i\right] = \min_{1\le j\le N}\left[\sum_{i=1}^{j-1} u(i,2) + \sum_{i=j+1}^{N} u(i,1)\right]. \end{aligned}$$

We have obtained the right equality in (18). Using (5), we get $r_i + d_i = a_i + s_{i+1} = u(i+1,1) + u(i+1,2)$. It follows that

$$R_N + D_N = \sum_{i=0}^{N-1} r_i + \sum_{i=0}^{N-1} d_i = \sum_{i=1}^{N} u(i,1) + \sum_{i=1}^{N} u(i,2) = |w(U)| = \lambda_1 + \lambda_2 .$$

Now according to (17), we have $\lambda_1 = D_N$ and we conclude that $\lambda_2 = R_N$. □

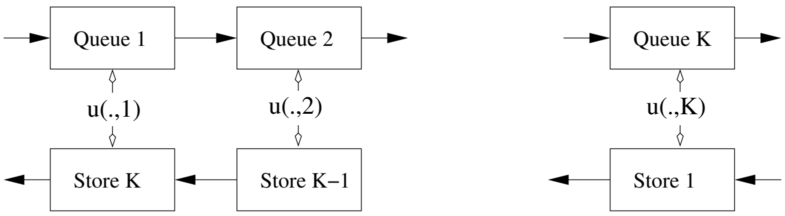

Fig. 6: Queues and stores are gone through in opposite orders

5.3 Statistical properties in the geometric case

If T is a tableau over the alphabet $\{1,\dots,k\}$, we write $x^T = \prod_{i=1}^k x_i^{\alpha_i}$, where α_i is the number of occurrences of the label i in the tableau. The *Schur function* s_λ associated with the partition $\lambda = (\lambda_1,\dots,\lambda_k)$ is defined by

$$s_\lambda(x_1,\dots,x_k) = \sum_{T:\ \mathrm{sh}(T)=\lambda} x^T ,$$

where $\mathrm{sh}(T)$ is the shape of the tableau T. We refer the reader to the books [21, 22] for more about Schur functions and their connection to the RSK correspondence.

Suppose that the random variables $(u(i,j))_{(i,j)\in\{1,\dots,N\}\times\{1,\dots,K\}}$ are independent and that $u(i,j)$ is geometrically distributed with parameter q_j, for some fixed $q = (q_1,\dots,q_K) \in (0,1)^K$. Then the law of the random partition $\lambda = (\lambda_1,\dots,\lambda_K)$ is given by $P\{\lambda = l\} = a(q)^N s_l(q) s_l(1,\dots,1), \quad l \in P_K$, where P_K is the set of integer partitions with at most K non-zero parts, $a(q) = \prod_j (1-q_j)$ and s_l is the Schur function associated with the integer partition l. In particular, the law of λ, and hence that of D and R, is *symmetric* in the parameters $q_1,\dots,q_K$. The fact that the law of D is symmetric in the parameters has been known in the queueing literature for some time [24, 1], but we see now that it also holds for the random variable R.

This extends to the level of *processes*: if we write $\lambda = \lambda^{(N)}$ to emphasize its dependence on N, it can be shown [14] that the random *sequence* $\lambda^{(N)}$ is a Markov chain in P_K with transition probabilities given by

$$P\{\lambda^{(N+1)} = l \,|\, \lambda^{(N)} = m\} = a(q)\frac{s_l(q)}{s_m(q)}$$

for all m and l such that $l_1 \geq m_1 \geq l_2 \geq m_2 \geq \cdots$. In particular, we can see that the law of the sequence $\lambda^{(N)}$, and hence also the laws of the sequences D_N and R_N, is symmetric in the parameters $q_1, \ldots, q_K$. Finally we remark that all of the above extends naturally to the exponential case.

References

[1] V. Anantharam. Probabilistic proof of the interchangeability of /M/1 queues in series. *Queueing Systems Theory Appl.*, 2(4):387–392, 1987.

[2] Y. Baryshnikov. GUEs and queues. *Probab. Theory Relat. Fields*, 119(2):256–274, 2001.

[3] A. Bedekar and M. Azizoğlu. The information-theoretic capacity of discrete-time queues. *IEEE Trans. Info. Theory*, 44(2):446–461, 1998.

[4] P. Brémaud. *Point Processes and Queues-Martingale Dynamics*. Springer Verlag, Berlin, 1981.

[5] P. Burke. The output of a queueing system. *Operations Research*, 4:699–704, 1956.

[6] D. Daley and D. Vere-Jones. *An Introduction to the Theory of Point Processes*. Springer Series in Statistics. Springer-Verlag, 1988.

[7] P. Glynn and W. Whitt. Departures from many queues in series. *Ann. Appl. Probab.*, 1(4):546–572, 1991.

[8] B. Hambly, J.. Martin, and N. O'Connell. Pitman's 2M-X theorem for skip-free random walks with Markovian increments. *Elect. Comm. in Probab.*, 6:73–77, 2001.

[9] J. Harrison and R. Williams. On the quasireversibility of a multiclass Brownian service station. *Ann. Probab.*, 18(3):1249–1268, 1990.

[10] F.P. Kelly. *Reversibility and Stochastic Networks*. Wiley, New-York, 1979.

[11] D. Knuth. Permutations, matrices, and generalized Young tableaux. *Pac. J. Math.*, 34:709–727, 1970.

[12] W. König, N. O'Connell, and S. Roch. Non-colliding random walks, tandem queues, and discrete orthogonal polynomial ensembles. *Elect. J. Probab.*, 7:1–24, 2002.

[13] E. Muth. The reversibility property of production lines. *Management science*, 25(2):152–158, 1979.

[14] N. O'Connell. Conditioned random walks and the RSK correspondence. *J. Phys. A*, 36:3049–3066, 2003.

[15] N. O'Connell. A path-transformation for random walks and the Robinson–Schensted correspondence. *Trans. Amer. Math. Soc.*, 355:3669–3697, 2003.

[16] N. O'Connell and M. Yor. Brownian analogues of Burke's theorem. *Stoch. Proc. Appl.*, 96(2):285–304, 2001.

[17] N. O'Connell and M. Yor. A representation for non-colliding random walks. *Elect. Comm. in Probab.*, 7:1–12, 2002.

[18] B. Prabhakar and R. Gallager. Entropy and the timing of discrete queues. *IEEE Transactions on Information Theory*, 2003. To appear.

[19] E. Reich. Waiting times when queues are in tandem. *Ann. Math. Stat.*, 28:527–530, 1957.

[20] P. Robert. *Réseaux et files d'attente: méthodes probabilistes*. Number 35 in Mathématiques & Applications. Springer, 2000.

[21] B. Sagan. *The symmetric group. Representations, combinatorial algorithms, and symmetric functions*. Number 203 in Graduate Texts in Mathematics. Springer-Verlag, 2001. 2nd edition.

[22] R. Stanley. *Enumerative Combinatorics, Vol. 2*. Number 62 in Cambridge Studies in Advanced Mathematics. Cambridge University Press, 1999.

[23] S. Tembe and R. Wolff. The optimal order of service in tandem queues. *Oper. Res.*, 24:824–832, 1974.

[24] R. Weber. The interchangeability of./M/1 queues in series. *J. Appl. Probab.*, 16(3):690–695, 1979.