

Stochastic queueing-theory approach to human dynamics

Joris Walraevens, Thomas Demoor, Tom Maertens, and Herwig Bruneel

Department of Telecommunications and Information Processing (EA07), Ghent University, B-9000 Ghent, Belgium

(Received 21 February 2011; revised manuscript received 13 December 2011; published 23 February 2012)

Recently, numerous studies have shown that human dynamics cannot be described accurately by exponential laws. For instance, Barabási [Nature (London) **435**, 207 (2005)] demonstrates that waiting times of tasks to be performed by a human are more suitably modeled by power laws. He presumes that these power laws are caused by a priority selection mechanism among the tasks. Priority models are well-developed in queueing theory (e.g., for telecommunication applications), and this paper demonstrates the (quasi-)immediate applicability of such a stochastic priority model to human dynamics. By calculating generating functions and by studying them in their dominant singularity, we prove that nonexponential tails result naturally. Contrary to popular belief, however, these are not necessarily triggered by the priority selection mechanism.

DOI: [10.1103/PhysRevE.85.021139](https://doi.org/10.1103/PhysRevE.85.021139)

PACS number(s): 02.50.-r, 89.65.Ef, 89.75.Da, 89.75.Fb

I. INTRODUCTION

Human dynamics is ubiquitous as human behavior has a strong impact on the dynamics of, for example, communication systems, (financial) business, health care services, and transport operations. We all perform “tasks,” be it at work or at home. Since multiple tasks interfere and cannot be performed simultaneously, the tasks of a human “queue up” while waiting for execution. Hence, the execution of tasks by a single human can be modeled as a queueing process.

Early models of human dynamics [1,2] assume that tasks arrive at the queue according to a Poisson process and that they are executed in a first-come-first-served (FCFS) order. This leads to an exponential distribution for the probability $D(n)$ that a task waits in the queue for n time periods, i.e., $D(n) \sim ce^{-\alpha n}$, for some c and α . Consecutive tasks (for instance, responding to two emails) are then executed at relatively regular time intervals, and very long waiting times occur with small probability. However, real-life measurements indicated that the timing of human actions is better approximated by a power-law distribution [$D(n) \sim cn^{-\alpha}$ for some c and α], allowing for long periods of inactivity between consecutive tasks alternated with bursts of intensive activity.

In a pioneering attempt to explain the observed power laws, Barabási [3] conjectures that human actions are driven by a priority selection mechanism. He assumes that the queue always contains a fixed number (L) of tasks, each being assigned a priority parameter x chosen from a random distribution $\rho(x)$. At each time step, the individual selects the highest-priority task with probability p , or, with probability $1 - p$, the individual randomly selects a task. All tasks have execution times of exactly one time period. After a task is executed, it is removed from the queue and a new task is added. Simulations with priorities chosen from a uniform distribution on $[0, 1]$ show that in the limit for $p \rightarrow 1$, thus when the highest priority task is always selected for execution, the waiting time distribution indeed has a power-law tail. The simulations also indicate that the tail of $D(n)$ is independent of L . In [4,5], Barabási’s model is therefore studied analytically for $L = 2$, with similar conclusions. Barabási thus presumes that the priority selection mechanism causes the observed power laws, a hypothesis that is copied in many subsequent, more advanced, models. The current paper demonstrates, among other things,

that although this hypothesis might be partially valid, it is not the only possible explanation for the power-law behavior.

Some of the subsequent models, furthermore, have dropped the unrealistic assumption that the queue contains the same number of tasks at any time, i.e., that each time a task is removed from the queue, a new task is added. Indeed, one can imagine that tasks arrive according to a time-varying process, with periods of many arriving tasks alternated with periods of few or no arrivals. Analogously, execution times of tasks are also potentially variable as some tasks might take a longer time to execute than others. So, in reality, the queue length fluctuates. Vázquez *et al.* [6] explore the behavior of priority queues in which there are no restrictions on the queue length. Tasks arrive according to a Poisson process with rate λ ; they are executed at rate μ . Each task is assigned a priority parameter x upon arrival. The highest priority task is always chosen for execution. A numerical study with priorities taking up continuous values from an arbitrary distribution on $[0, \infty)$ shows that for $0 < \lambda < \mu$, the waiting time distribution is characterized by a power-law decay combined with an exponential cutoff [$D(n) \sim cn^{-\alpha_1} e^{-\alpha_2 n}$ for some c, α_1 , and α_2]. For $\lambda \geq \mu$, a pure power-law distribution emerges, but only for the tasks that are executed (tasks with low priority remain in the queue forever). This model is analyzed probabilistically in Ref. [7], yielding similar conclusions. Furthermore, Masuda *et al.* [8] analytically study a priority queue where the number of incoming tasks follows a power-law distribution. They show that the waiting time exhibits a power-law behavior with α ’s that depend on the parameters of the arrival process.

In this paper, we propose a general stochastic framework that encapsulates the results of Refs. [6,8] as special cases, and also leads to new results and insights. We assume that the numbers of arrivals per time step and the execution times of tasks are discrete stochastic variables [9], with distributions that can be chosen arbitrarily, and that the priority levels are uniformly distributed on $[0, 1]$. The modeling assumptions are detailed in the next section. Regarding modeling issues, the merit of the current contribution lies not only in the more general model but especially in embodying that human dynamics models can be generated from (priority) queueing models that are readily available in queueing-theory literature. The main complexity here is the transformation of continuously many

priority levels (in human dynamics) to a discrete number of priority classes (that is generally assumed in queueing theory). In this paper, we use the discrete-time two-class preemptive priority model of Ref. [10] to study the sojourn time (i.e., waiting time plus execution time) distribution in human dynamics. In Ref. [10], we make extensive use of generating functions, so we first demonstrate in Sec. III that calculating the generating function of the variable of interest and performing singularity analysis on this generating function, which is quite standard in queueing theory and analytic combinatorics (see, e.g., [11]), is an accurate and viable technique for predicting tail behavior. The actual analysis of the steady-state sojourn time distribution of a task with given priority x is performed in Sec. IV. By means of several instructive examples, where we focus on the tail behavior, we demonstrate that power laws do not always show up in a priority model, and that, when they do, they are not necessarily triggered by the priority selection mechanism. Finally, the paper is concluded in Sec. V, which also touches upon possible future research directions.

II. THE MODEL

In each discrete time step, n tasks arrive with given probability mass function $a(n)$ ($n = 0, 1, 2, \dots$). We define the corresponding generating function as $\Lambda(z)$. The numbers of arrivals in two different time steps are independent. Each arriving task is assigned a priority level x , with x drawn from a uniform distribution on $[0, 1]$. When tasks are present in the queue at the beginning of a given time step, the task with highest priority (highest x) is selected for execution in that time step. The total execution time of a random task takes n time steps with probability $s(n)$ ($n = 1, 2, \dots$) and the corresponding generating function is indicated by $S(z)$. This model is a generalization of the model adopted in Ref. [8], where the authors assume a specific power-law distribution for the number of arriving tasks in a time step and a shifted geometric distribution for the execution times of tasks. In Ref. [6], a continuous-time model is assumed, where the arrival process is modeled as a Poisson process and the execution times are exponentially distributed. We, on the contrary, do not consider specific distributions for the number of arriving tasks nor for the execution times, since this is not mandatory for the analysis.

Note that the average work that is brought into the system during a time step is assumed to be smaller than what can be executed in a time step, i.e., the average number of arriving tasks multiplied by the average execution time is supposed to be strictly less than 1. This is a necessary condition for the system to reach a steady state (stability condition). This stability condition is not fulfilled in the models of Refs. [3,4], which raises issues that we will discuss in Sec. IV C 1. First, let us justify the use of generating functions.

III. THE GENERATING-FUNCTION APPROACH

The generating function $Y(z)$ of a discrete random variable y is the z transform of its probability mass function $y(n)$:

$$Y(z) = \sum_{n=0}^{\infty} y(n)z^n,$$

with z being a complex number and $Y(1) = 1$. Generating functions are an elegant tool for manipulating discrete functions and discrete random variables. They further incorporate all information about the moments of the variable and about the tail of the distribution. The moments are calculated by invoking the moment-generating property:

Theorem 1. Let $Y(z)$ be the generating function of a random variable y . Then

$$E \left[\prod_{i=0}^{n-1} (y - i) \right] = Y^{(n)}(1),$$

with $Y^{(n)}$ the n th derivative of Y .

For instance, the expected value $E[y]$ is given by $Y'(1)$.

The tail of the distribution at hand is perhaps even more interesting. This tail is *completely* characterized by the lowest-norm singularity R_Y of $Y(z)$ (the so-called dominant singularity [12]) and the behavior of $Y(z)$ in the neighbourhood of this singularity. It is generally known that $R_Y \in [1, \infty]$. For $z \in [0, R_Y]$, $Y(z)$ is a positive-real strictly increasing function, so the inverse function $Y^{-1}(z)$ can be defined on the real interval $[Y(0), Y(R_Y)]$.

The following theorem (see [11]) enables a characterization of the tail of the distribution of a random variable y from its generating function:

Theorem 2. Let $Y(z)$ be the generating function of a random variable y , with dominant singularity R_Y . Let $\beta \in \mathbb{R} \setminus \{0, 1, 2, \dots\}$. If for $z \rightarrow R_Y$

$$Y(z) \sim c_Y (1 - z/R_Y)^\beta,$$

then the distribution $y(n)$ satisfies

$$y(n) \sim \frac{c_Y n^{-\beta-1} R_Y^{-n}}{\Gamma(-\beta)}$$

for $n \rightarrow \infty$, with $\Gamma(\dots)$ the Gamma function.

Basically, if the dominant singularity of a generating function (R_Y) and the behavior of the generating function in the neighbourhood of this dominant singularity (β and c_Y) are identified, this theorem expresses the tail of the corresponding distribution [$y(n)$ for large n]. A dominant pole of multiplicity 1 ($\beta = -1$) in the interval $]1, \infty[$, for example, leads to an exponential tail. If the dominant singularity is 1 ($R_Y = 1$), a power-law tail is encountered.

IV. ANALYSIS OF SOJOURN TIME

A. Mapping and analysis

The sojourn time of a task is defined as the number of time steps between the end of the time step when the task arrives and the instant the task is fully executed. Randomly tag a certain task and assume its priority level equals x . Let d_x indicate its sojourn time and $D_x(z)$ be the corresponding generating function. This task's sojourn time is only affected by tasks of higher priority, as all tasks of lower priority are executed after the tagged task. Moreover, only the total number of tasks of higher priority, and not the order in which these tasks are executed, influences d_x . Consequently, we can map the current model with continuously many priority levels onto a general two-class priority model from the queueing literature [10]. Here, tasks are categorized into two distinct classes, high

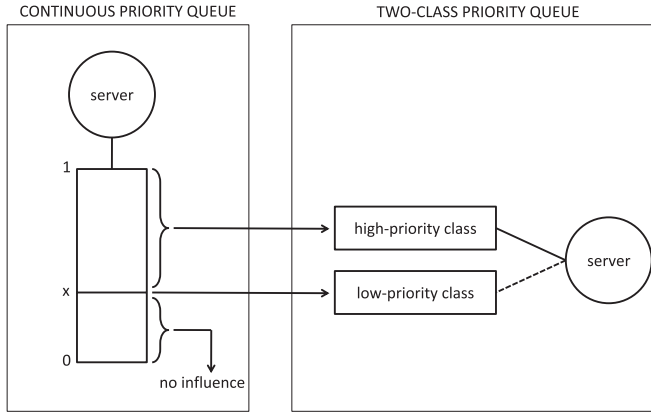


FIG. 1. Mapping of continuous priority queue to two-class priority queue.

and low priority, and the server only processes tasks of low priority when there are no high-priority tasks waiting. Thus, if we assign the task with priority level x to the low-priority class and aggregate all tasks of higher priority level into the high-priority class (as the order in which they are executed is irrelevant), the sojourn time of a low-priority task is identical to d_x . This mapping is depicted in Fig. 1. Note that the fact that the low-priority task is alone in its class (the probability of a task having priority level exactly equal to x is zero) poses a small issue, which is resolved below.

In the two-class priority model [10], the generating function of the sojourn time of a low-priority task is given by

$$D(z) = \frac{\mu - \lambda_T}{(\lambda_T - \lambda_H)\mu} \frac{V(z)\{zA_H[V(z)] - 1\}}{A_H[V(z)][V(z) - 1]} \frac{A_T[V(z)] - A_H[V(z)]}{zA_H[V(z)] - A_T[V(z)]}, \quad (1)$$

with $A_T(z)$ [$A_H(z)$, respectively] the generating function of the total number of tasks (number of high-priority tasks, respectively) arriving in the queue in a time step, $\lambda_T = A'_T(1)$ [$\lambda_H = A'_H(1)$, respectively] the corresponding means, $S(z)$ the generating function of the execution times, $\mu = 1/S'(1)$ the execution rate, and $V(z) = S\{zA_H[V(z)]\}$, with $|V(z)| < 1$ for $|z| < 1$. We note that $V(z)$ is implicitly (self-consistently) yet uniquely defined and is, in fact, a generating function.

Let us now complete the mapping by translating expression (1) to the original model. This boils down to the calculation of $A_H(z)$ and $A_T(z)$ and the corresponding means λ_H and λ_T . As high-priority tasks are tasks with a priority level higher than x and as the probability that the priority level of an arriving task is larger than x equals $1 - x$, $A_H(z)$ equals

$$\begin{aligned} \Lambda_x(z) &= \sum_{l=0}^{\infty} \sum_{n=l}^{\infty} a(n) \binom{n}{l} (1-x)^l x^{n-l} z^l \\ &= \sum_{n=0}^{\infty} a(n) [(1-x)z + x]^n \\ &= \Lambda[(1-x)z + x]. \end{aligned}$$

Since the low-priority class contains the tagged task only, $A_T(z) = A_H(z)$ (and, hence, $\lambda_T = \lambda_H$). In Eq. (1), this leads to a numerator and denominator equal to zero. This complication

is tackled by assuming that, instead of having only a single low-priority task, all tasks with a priority level in $]x - \epsilon, x]$ are part of the low-priority class [i.e., $A_T(z) = \Lambda_{x-\epsilon}(z)$] and subsequently taking the limit $\epsilon \rightarrow 0$. This yields

$$\begin{aligned} D_x(z) &= \lim_{\epsilon \rightarrow 0} \frac{\mu - \lambda(1-x+\epsilon)}{\lambda\epsilon\mu} \frac{V_x(z)\{z\Lambda_x[V_x(z)] - 1\}}{\Lambda_x[V_x(z)][V_x(z) - 1]} \\ &\quad \frac{\Lambda_{x-\epsilon}[V_x(z)] - \Lambda_x[V_x(z)]}{z\Lambda_x[V_x(z)] - \Lambda_{x-\epsilon}[V_x(z)]} \\ &= \frac{\mu - \lambda_x}{\lambda_x\mu} \frac{V_x(z)\{z\Lambda_x[V_x(z)] - 1\}}{z - 1} \frac{\Lambda'_x[V_x(z)]}{\Lambda_x[V_x(z)]^2}, \quad (2) \end{aligned}$$

with $\lambda = \Lambda'(1)$ the arrival rate of tasks, $\lambda_x = \Lambda'_x(1) = (1-x)\lambda$ the arrival rate of tasks with priority level higher than x , and $V_x(z)$ implicitly defined as

$$V_x(z) = S\{z\Lambda_x[V_x(z)]\}, \quad (3)$$

with $|V_x(z)| < 1$ for $|z| < 1$. Expression (2) contains all information about the sojourn time of a task with priority level x . Next, we immediately obtain the expected sojourn time of a task with priority level x and of a random task (irrespective of its priority level) and show how singularity analysis of Eq. (2) leads to asymptotics.

B. Expected sojourn time

The expected sojourn time of a task with priority level x is given by

$$\begin{aligned} E[d_x] &= D'_x(1) \\ &= \frac{(2\mu - \lambda_x)\text{var}[\Lambda_x]}{2\lambda_x(\mu - \lambda_x)^2} + \frac{\lambda_x\mu^2\text{var}[S]}{2(\mu - \lambda_x)^2} - \frac{\lambda_x}{2(\mu - \lambda_x)}, \end{aligned}$$

with $\text{var}[\Lambda_x]$ and $\text{var}[S]$ the variances of the number of task arrivals in a time step with priority level higher than x and of the execution times, respectively.

By integrating over x , we find the mean sojourn time of a random task:

$$\begin{aligned} E[d] &= \int_0^1 E[d_x] dx \\ &= \frac{\text{var}[\Lambda]}{2\lambda(\mu - \lambda)} + \frac{\mu^2[\lambda - (\mu - \lambda)\ln(1 - \lambda/\mu)]\text{var}[S]}{2\lambda(\mu - \lambda)} \\ &\quad + \frac{\lambda + (1 - \mu)\ln(1 - \lambda/\mu)}{2\lambda}, \end{aligned}$$

with $\text{var}[\Lambda]$ the variance of the total number of arriving tasks in a time step.

C. Tail asymptotics of the sojourn time

Characterizing the tail of the distribution of d_x first requires discovering the dominant singularity R_{D_x} of $D_x(z)$. By investigating expression (2), it is clear that $R_{D_x} = \min(R_{V_x}, V_x^{-1}(R_{\Lambda_x}))$, since R_{V_x} and $V_x^{-1}(R_{\Lambda_x})$ are the dominant singularities of $V_x(z)$ and $\Lambda_x[V_x(z)]$ [13]. The input functions $\Lambda(z)$ and $S(z)$ determine which of the two singularities is dominant. Once this dominant singularity is determined, we investigate the behavior of $D_x(z)$ in its neighbourhood. Finally, invoking theorem 2 delivers the tail behavior of $d_x(n)$.

Since writing down the singularity analysis for all possible types of input functions $\Lambda(z)$ and $S(z)$ is a tedious task,

we demonstrate the procedure by means of four thoroughly worked-out, instructive examples. They represent important classes of human dynamics. We stress that the discussed cases are just examples of the power of our approach. The procedure works equally well for any other execution time distribution and/or distribution of the number of incoming tasks: just perform singularity analysis on Eq. (2) and apply theorem 2.

Before initiating the examples, we remark that the sojourn time of a *random* task that is effectively executed, irrespective of its priority level, is dominated by the sojourn time of the lowest priority task that is effectively executed [6], i.e., its probability mass function $d(n) \sim \lim_{x \rightarrow x^*} d_x(n)$ with $x^* = \max(1 - \mu/\lambda, 0)$.

1. Geometric input

The first example is closely related to Ref. [6]. Consider geometric distributions for the numbers of arrivals in a time step as well as for the execution times, i.e.,

$$\Lambda(z) = \frac{1}{1 + \lambda - \lambda z}, \quad (4)$$

$$S(z) = \frac{\mu z}{1 - (1 - \mu)z}. \quad (5)$$

We first investigate $V_x(z)$ and R_{V_x} . In this particular case, $V_x(z)$ can be calculated explicitly:

$$V_x(z) = \frac{1 + \lambda_x - (1 - \mu)z}{2\lambda_x} - \sqrt{\left(\frac{1 + \lambda_x - (1 - \mu)z}{2\lambda_x}\right)^2 - \mu z}.$$

From this expression, it is easy to see that the only singularities are the zeros of the radicand (part under the square root). The dominant one equals

$$R_{V_x} = \frac{1 - \mu(1 - \lambda_x) + \lambda_x - 2\sqrt{\lambda_x \mu(1 - \mu + \lambda_x)}}{(1 - \mu)^2}. \quad (6)$$

This dominant singularity is a square-root-like branch point [14], which is typical for priority queueing systems (see, e.g., [10, 15]). It can further be proved that $R_{V_x} < V_x^{-1}(R_{\Lambda_x})$, so the dominant singularity of $D_x(z)$, R_{D_x} , is found to be R_{V_x} .

Substituting the particular expressions for $\Lambda(z)$, $S(z)$, and $V_x(z)$ in expression (2) yields

$$D_x(z) = D_x(R_{V_x}) - c_{V_x}(1 - z/R_{V_x})^{1/2}$$

with

$$c_{V_x} = \frac{(\mu - \lambda_x) \sqrt{(1 + \lambda_x)^2 - R_{V_x}^2(1 - \mu)^2}}{2\lambda_x(1 - \mu + \lambda_x)(R_{V_x} - 1)}.$$

Then, application of theorem 2 leads to

$$d_x(n) \sim \frac{c_{V_x}}{2\sqrt{\pi}} n^{-3/2} R_{V_x}^{-n}. \quad (7)$$

We do not obtain a pure power law ($\sim cn^{-\alpha}$) in this case, as conjectured in Ref. [3], nor an exponential law ($\sim c\tau^{-n}$), but a so-called power law with an exponential cutoff ($\sim cn^{-\alpha} \tau^{-n}$). This is typical for priority queueing systems with “standard” input (basically, input generating functions that have

a dominant *pole* or are analytic in the whole complex plane; see [10, 15], but also the model of Ref. [6] belongs to this category), as is the case for the geometric input in this example. Here, the tail behavior is thus caused by the priority selection mechanism.

The reason why power laws are encountered in Refs. [3, 4] is that their queueing models are on the borderline of instability (the amount of arriving work is exactly equal to the amount of work that can be executed). In Ref. [7], it is proved that such instable models indeed lead to power laws. This also emerges from our results: if λ_x approaches μ , R_{V_x} goes to 1 in the limit [see expression (6)], so the exponential cutoff vanishes [factor $R_{V_x}^{-n}$ in Eq. (7)]. So when $\lambda \geq \mu$, we find that a random task that is executed has a power-law tail (see also [6]).

2. Power-law number of arrivals

As a second example, we look at the model of Ref. [8]. Execution times are geometrically distributed according to formula (5), whereas the number of arrivals in a time step has a power-law distribution, i.e., $a(0) = 1 - \alpha$ and

$$a(n) = \alpha \frac{n^{-\gamma}}{\text{Li}_\gamma(1)}, \quad n > 0, \quad (8)$$

with $\text{Li}_\gamma(z) = \sum_{i=1}^{\infty} i^{-\gamma} z^i$ the so-called polylogarithm. We assume $\gamma \in]2, 3[$ for expository reasons [16]. Then, the average number of arrivals in a time step,

$$\lambda = \alpha \frac{\text{Li}_{\gamma-1}(1)}{\text{Li}_\gamma(1)},$$

is finite but the variance is infinite. Therefore, the corresponding generating function

$$\Lambda(z) = 1 - \alpha + \alpha \frac{\text{Li}_\gamma(z)}{\text{Li}_\gamma(1)}$$

has a branch point in $z = 1$ and can be written as

$$\Lambda(z) \sim 1 - \lambda(1 - z) + c_\Lambda(1 - z)^{\gamma-1},$$

with $c_\Lambda = \alpha\Gamma(1 - \gamma)/\text{Li}_\gamma(1)$.

It can be proven that the branch point in $z = 1$ of $\Lambda(z)$ is transferred to $V_x(z)$; $V_x(z)$ can be written as

$$V_x(z) \sim 1 - \frac{1}{\mu - \lambda_x}(1 - z) + \frac{c_\Lambda(1 - x)^{\gamma-1}}{(\mu - \lambda_x)^\gamma}(1 - z)^{\gamma-1}.$$

Hence, R_{V_x} and $V_x^{-1}(R_{\Lambda_x})$ are both equal to 1.

Consequently, $R_{D_x} = 1$ and $D_x(z)$ can be formulated by

$$D_x(z) \sim 1 - \frac{c_\Lambda(1 - x)^{\gamma-1}[\lambda_x + (\mu - \lambda_x)(\gamma - 1)]}{\lambda_x(\mu - \lambda_x)^{\gamma-1}} \times (1 - z)^{\gamma-2}.$$

Using theorem 2, we find

$$d_x(n) \sim \frac{(1 - x)^{\gamma-2}[\lambda_x + (\mu - \lambda_x)(\gamma - 1)]}{(\gamma - 1)(\mu - \lambda_x)^{\gamma-1} \text{Li}_{\gamma-1}(1)} n^{-\gamma+1}. \quad (9)$$

We note that this tail is a power law that is one degree heavier than that of the input function $a(n)$. This is a general result for queueing systems with power-law input, not only in queueing systems with a priority selection mechanism but also in queueing systems with a FCFS or random selection protocol.

For instance, for the model described in this paragraph, but with a FCFS scheduling instead of a priority scheduling, we find (see, for instance, [17])

$$D_{\text{FCFS}}(z) = \frac{\mu - \lambda_T}{\lambda_T} \frac{z\{\Lambda[S(z)] - 1\}}{z - \Lambda[S(z)]} \sim 1 - \frac{c_\Lambda}{\lambda_T \mu^{\gamma-3} (\mu - \lambda_T)} (1-z)^{\gamma-2},$$

and, by means of theorem 2,

$$d_{\text{FCFS}}(n) \sim \frac{1}{(\gamma - 1)\mu^{\gamma-3}(\mu - \lambda_T)\text{Li}_{\gamma-1}(1)} n^{-\gamma+1}.$$

For a model with a random order of service, we refer the reader to Ref. [18].

The power law reported in Ref. [8] is thus not a result of priority selection of tasks, as it also occurs with other selection policies, but rather of the power law of one of the input functions [the branch point in $z = 1$ of the input function $\Lambda(z)$ is “transferred” to the output function $D_x(z)$]. This seems to be a novel observation in human dynamics models and studies.

3. Power-law execution times

In all models in the literature, execution times are assumed to be geometric or exponential. However, one can imagine that, in reality, execution times might have a high(er) variability. Therefore, in this example, we assume a power-law distribution for the execution times,

$$S(z) = \frac{\text{Li}_\gamma(z)}{\text{Li}_\gamma(1)},$$

and $\mu = \text{Li}_\gamma(1)/\text{Li}_{\gamma-1}(1)$. Again, let us assume that $\gamma \in]2,3[$. $S(z)$ has a branch point in 1 and thus

$$S(z) \sim 1 - \frac{1}{\mu}(1-z) + c_S(1-z)^{\gamma-1},$$

with $c_S = \Gamma(1-\gamma)/\text{Li}_\gamma(1)$. Furthermore, let the number of arrivals in a time step be geometrically distributed as in Eq. (4).

The branch point of $S(z)$ in $z = 1$ is transferred to $V_x(z)$ as in the previous example:

$$V_x(z) \sim 1 - \frac{1}{\mu - \lambda_x}(1-z) + \frac{c_S \mu^\gamma}{(\mu - \lambda_x)^\gamma} (1-z)^{\gamma-1}.$$

Then, expression (2) for $D_x(z)$ becomes

$$D_x(z) \sim 1 - \frac{c_S \lambda_x \mu^{\gamma-1}}{(\mu - \lambda_x)^{\gamma-1}} (1-z)^{\gamma-2},$$

and further, by using theorem 2, we discover

$$d_x(n) \sim \frac{\lambda_x \mu^{\gamma-1}}{(\gamma - 1)(\mu - \lambda_x)^{\gamma-1} \text{Li}_\gamma(1)} n^{-\gamma+1}. \quad (10)$$

The tail of the delay is again a power law one degree heavier than that of the power-law input function, in this case the execution times. It is easy to check that, analogous to the previous example, a FCFS selection mechanism leads to the same behavior. The observed power law is thus again not caused by priority selection but rather is the direct result of the power law in the execution times.

4. Power-law number of arrivals with an exponential cutoff

In the previous three examples, the tail of the execution times is either determined by the priority selection (example 1) or by an input function (examples 2 and 3). As a final example, we treat a case in which both effects arise. Here, the parameters of the input functions govern which of both effects determines the tail of the sojourn time. We will not go into full details of the singularity analysis, but merely touch upon relevant modeling issues and results.

Execution times are geometrically distributed according to formula (5), whereas the number of arrivals in a time step has distribution $a(0) = 1 - \alpha$ and

$$a(n) = \alpha \frac{n^{-\gamma} \tau^{-n}}{\text{Li}_\gamma(1/\tau)}, \quad n > 0.$$

Note that $\tau = 1$ leads to the power law in Eq. (8). Let us assume here that $\tau > 1$, leading to a power law with an exponential cutoff. As in example 2, we consider $\gamma \in]2,3[$. The average number of arrivals in a time step equals

$$\lambda = \alpha \frac{\text{Li}_{\gamma-1}(1/\tau)}{\text{Li}_\gamma(1/\tau)}.$$

The corresponding generating function

$$\Lambda(z) = 1 - \alpha + \alpha \frac{\text{Li}_\gamma(z/\tau)}{\text{Li}_\gamma(1/\tau)}$$

has a branch point in $z = \tau > 1$.

Two singularities of $V_x(z)$ and $D_x(z)$ can be dominant. The first singularity is the one that originates directly from the input function’s branch point τ . This singularity can be transferred to the output function through the definition of $V_x(z)$ [expression (3)]. It is given by

$$R_{D_x,1} = \frac{S^{-1}(\tau_x)}{\Lambda_x(\tau_x)},$$

with $\tau_x = (\tau - x)/(1 - x)$ the branch point of $\Lambda_x(z)$. When $R_{D_x,1}$ is the dominant singularity of $D_x(z)$, this results in a power-law with an exponential cutoff ($\sim c_{D_x,1} n^{-\gamma+1} R_{D_x,1}^{-n}$).

A second singularity of $D_x(z)$ that might appear is the “priority” square-root branch point, also encountered in the first example. This square-root branch point leads to $V'_x(z) = \infty$. In this case, it turns out that this singularity cannot be calculated explicitly, but it can be calculated numerically from

$$R_{D_x,2} = \frac{V_x}{[\mu + (1 - \mu)V_x]\Lambda_x(V_x)},$$

with V_x the smallest solution in $[1, \infty]$ of

$$\frac{(1 - \mu)V_x}{\mu + (1 - \mu)V_x} + \frac{V_x \Lambda'_x(V_x)}{\Lambda_x(V_x)} = 1.$$

When $R_{D_x,2}$ is the dominant singularity of $D_x(z)$, this results in a power-law with an exponential cutoff ($\sim c_{D_x,2} n^{-3/2} R_{D_x,2}^{-n}$), as in the first example [see expression (7)].

Now, it remains to be discovered which of both singularities is dominant ($R_{D_x,1}$ or $R_{D_x,2}$). This depends on the parameters of the system. Basically, it can be expressed in terms of λ_x : if $\lambda_x < \lambda_x^*$, $R_{D_x,1}$ is dominant, and if $\lambda_x > \lambda_x^*$, $R_{D_x,2}$ is dominant, where λ_x^* can be expressed in terms of the input parameters of the system. Thus, for small λ_x , the priority effect is minimal

and the tail is determined by the branch point $R_{D_x,1}$ originating from the input function. For increasing λ_x , more tasks arrive and the priority effect kicks in. From λ_x^* onwards, the tail of the execution times is determined by the branch point $R_{D_x,2}$ caused by the priority scheduling. Note that, for the power-law case $\tau = 1$, $\lambda_x^* = \mu$ [the (in)stability border] and in stable systems ($\lambda_x < \mu = \lambda_x^*$) the tail is therefore *always* determined by the input function, as described in example 2.

V. CONCLUSIONS AND OUTLOOK

We have considered a general model for priority selection in human dynamics. We show that existing models in the queueing literature can easily be used in a human-dynamics context by mapping the continuously many priority levels to a discrete number of priority classes. Furthermore, we have demonstrated the power of singularity analysis of generating functions for obtaining tail asymptotics of relevant distributions in such models. Our main conclusions are that nonexponential laws may emerge in such models, yet pure power laws are not caused by priority selection (except in unstable systems) but by power-law input functions. However, it is established that priority selection may, depending on the input, give rise to power laws with an exponential cutoff. Apart from proving that tail behavior can be carried over from the input functions to the sojourn time, we also show that for certain types of input functions, tail behavior can shift between different types when the parameters of the input functions are varied.

This paper focused on a particular model to demonstrate the power of our analysis tools. However, many interesting models

could be investigated using this technique, starting from other existing queueing models. Hence, we conclude this paper with some possible alterations; per alternative model, some pointers to related queueing-theory papers are given, which could serve as a starting point. First, instead of the studied preemptive model, we could look at a non-preemptive model (once the execution of a task has started, it is not interrupted even if tasks with higher priority arrive at the queue during the process), or a mixture of both. For potential models and analyses, we refer the reader to Refs. [19,20]. Another interesting model arises when the priority level of a task may depend on its execution time: that is, people are inclined to give higher priority to easy (short) tasks. A model where the tasks with the shortest execution times are selected first is called the shortest job first selection or the variant shortest remaining processing time (see [21]). Yet another interesting complication could be modeling the evolution of the priority level of waiting tasks: the priority level of tasks might, for instance, increase during their time in the queue (see, e.g., [22,23]). The pure priority selection mechanism can also be altered, e.g., to a hybrid priority-FCFS or priority-random selection mechanism. Finally, some kind of coupling between different queues is an important topic, for instance for modeling the interaction between (the tasks of) two people [24–26]. An analysis of a useful coupled model can be found in Ref. [27].

ACKNOWLEDGMENTS

J.W. acknowledges the support of the Research Foundation–Flanders (FWO–Vlaanderen).

-
- [1] F. Haight, *Handbook of the Poisson Distribution* (Wiley, New York, 1967).
- [2] P. Reynolds, *Call Center Staffing* (The Call Center School Press, Lebanon, TN, 2003).
- [3] A.-L. Barabási, *Nature (London)* **435**, 207 (2005).
- [4] A. Vázquez, *Phys. Rev. Lett.* **95**, 248701 (2005).
- [5] A. Gabrielli and G. Caldarelli, *Phys. Rev. Lett.* **98**, 208701 (2007).
- [6] A. Vázquez, J. G. Oliveira, Z. Dezső, K.-I. Goh, I. Kondor, and A.-L. Barabási, *Phys. Rev. E* **73**, 036127 (2006).
- [7] G. Grinstein and R. Linsker, *Phys. Rev. E* **77**, 012101 (2008).
- [8] N. Masuda, J. S. Kim, and B. Kahng, *Phys. Rev. E* **79**, 036106 (2009).
- [9] A discrete variable y is a variable with a discrete state space, usually the non-negative numbers. It can be fully characterized by its probability mass function $y(n) = P[y = n]$, $n = 0, 1, 2, \dots$ (with P a probability measure), or, equivalently, by its generating function $Y(z) = E[z^y] = \sum_{n=0}^{\infty} y(n)z^n$ (with E the expectation operator).
- [10] J. Walraevens, B. Steyaert, and H. Bruneel, *Eur. J. Oper. Res.* **186**, 182 (2008).
- [11] P. Flajolet and R. Sedgewick, *Analytic Combinatorics* (Cambridge University Press, New York, 2008).
- [12] A singularity of a complex function is a point in the complex plane where the function is not analytic. A pole is the prime example, but a square-root branch point and branch cut are also singularities.
- [13] We note that the factor $z - 1$ in the denominator of Eq. (2) leads to a removable singularity, as required [$D_x(1) = 1$]; the numerator also equals 0 for $z = 1$.
- [14] In this specific case, this square-root branch point can be calculated explicitly. In most cases, however, a numerical procedure is necessary to calculate this branch point as this is the point where the first derivative of $V_x(z)$ is infinite.
- [15] J. Abate and W. Whitt, *Queueing Syst.* **25**, 173 (1997).
- [16] The final formula (9) is correct for all nonintegers $\gamma > 2$, but the intermediate formulas are somewhat different.
- [17] H. Bruneel and B. Kim, *Discrete-time Models for Communication Systems Including ATM* (Kluwer Academic, Boston, 1993).
- [18] O. Boxma, S. Foss, J.-M. Lasgouttes, and R. Núñez Queija, *Queueing Syst.* **46**, 35 (2004).
- [19] J. Walraevens, B. Steyaert, M. Moeneclaey, and H. Bruneel, *Telecommun. Syst.* **30**, 81 (2005).
- [20] K. Kim and K. Chae, *Eur. J. Oper. Res.* **200**, 473 (2010).
- [21] M. Nuyens and B. Zwart, *Queueing Syst.* **54**, 85 (2006).
- [22] P. Blanchard and M.-O. Hongler, *Phys. Rev. E* **75**, 026102 (2007).
- [23] T. Maertens, J. Walraevens, and H. Bruneel, *Performance Eval.* **63**, 1235 (2006).

- [24] J. Oliveira and A. Vazquez, *Physica A* **388**, 187 (2009).
- [25] W.-K. Cho, B. Min, K.-I. Goh, and I.-M. Kim, *Phys. Rev. E* **81**, 066109 (2010).
- [26] Y. Wu, C. Zhou, J. Xiao, J. Kurths, and H. Schellnhuber, *Proc. Natl. Acad. Sci. (USA)* **107**, 18803 (2010).
- [27] F. Guillemin and D. Pinchon, *J. Appl. Probab.* **41**, 832 (2004).